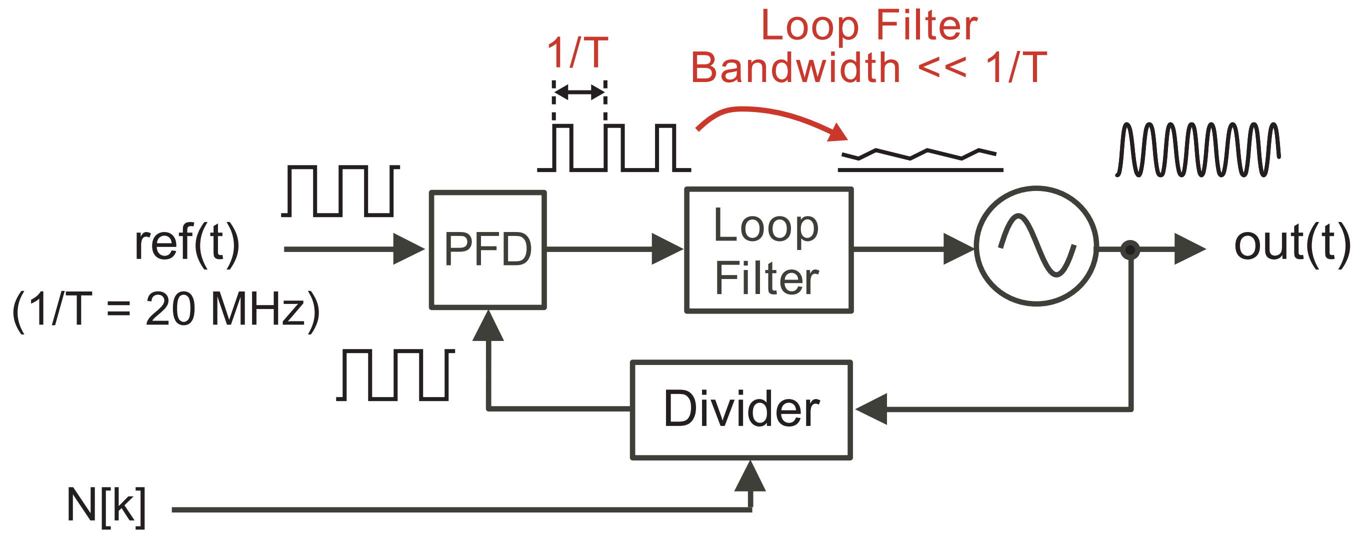
6.976
High Speed Communication Circuits and Systems
Lecture 17
Advanced Frequency Synthesizers

Michael Perrott

Massachusetts Institute of Technology

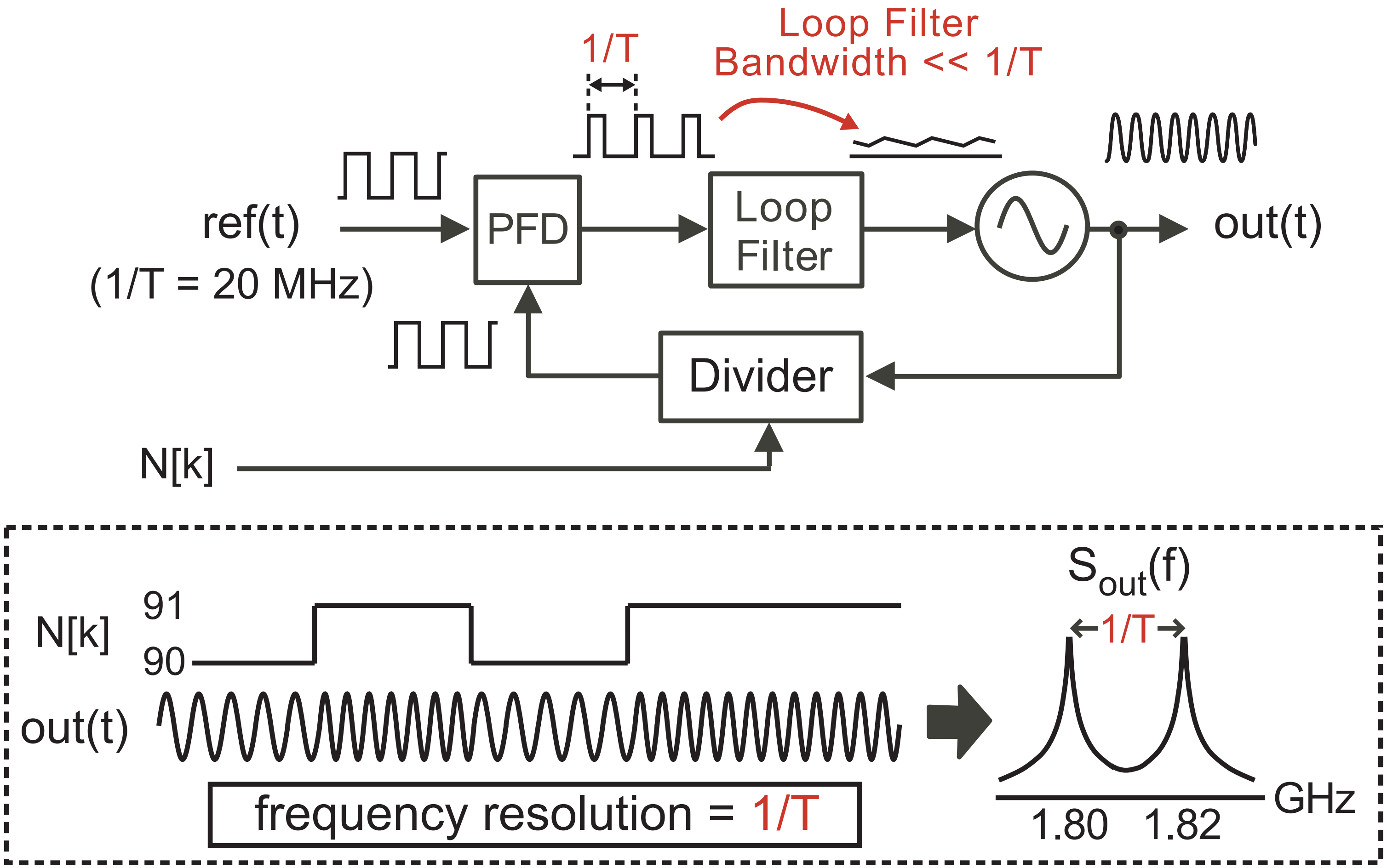
Bandwidth Constraints for Integer-N Synthesizers



- **PFD output has a periodicity of 1/T**
  - **1/T = reference frequency**
- **Loop filter must have a bandwidth << 1/T**
  - **PFD output pulses must be filtered out and average value extracted**

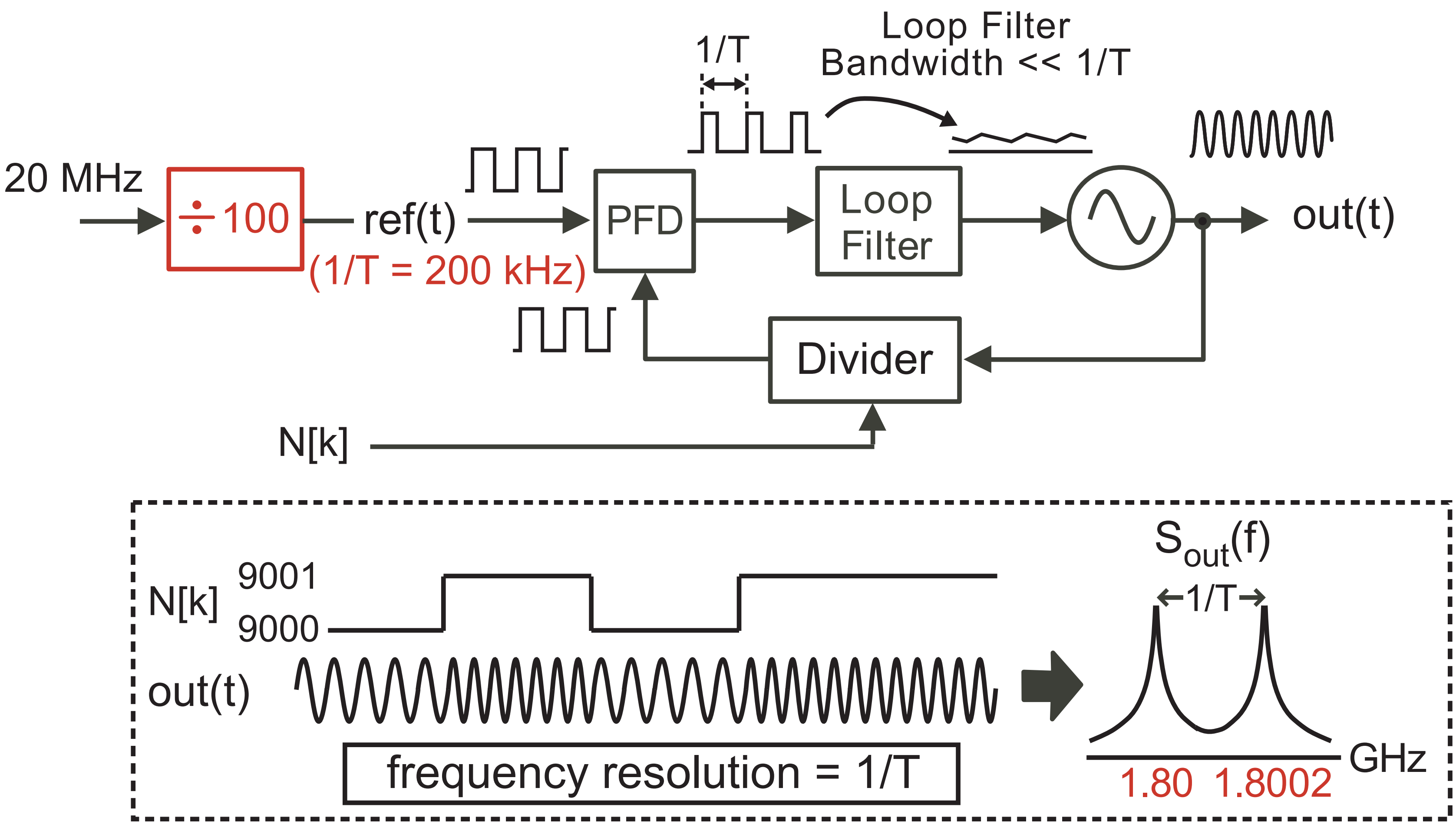
Closed loop PLL bandwidth often chosen to be a factor of ten lower than 1/T

Bandwidth Versus Frequency Resolution



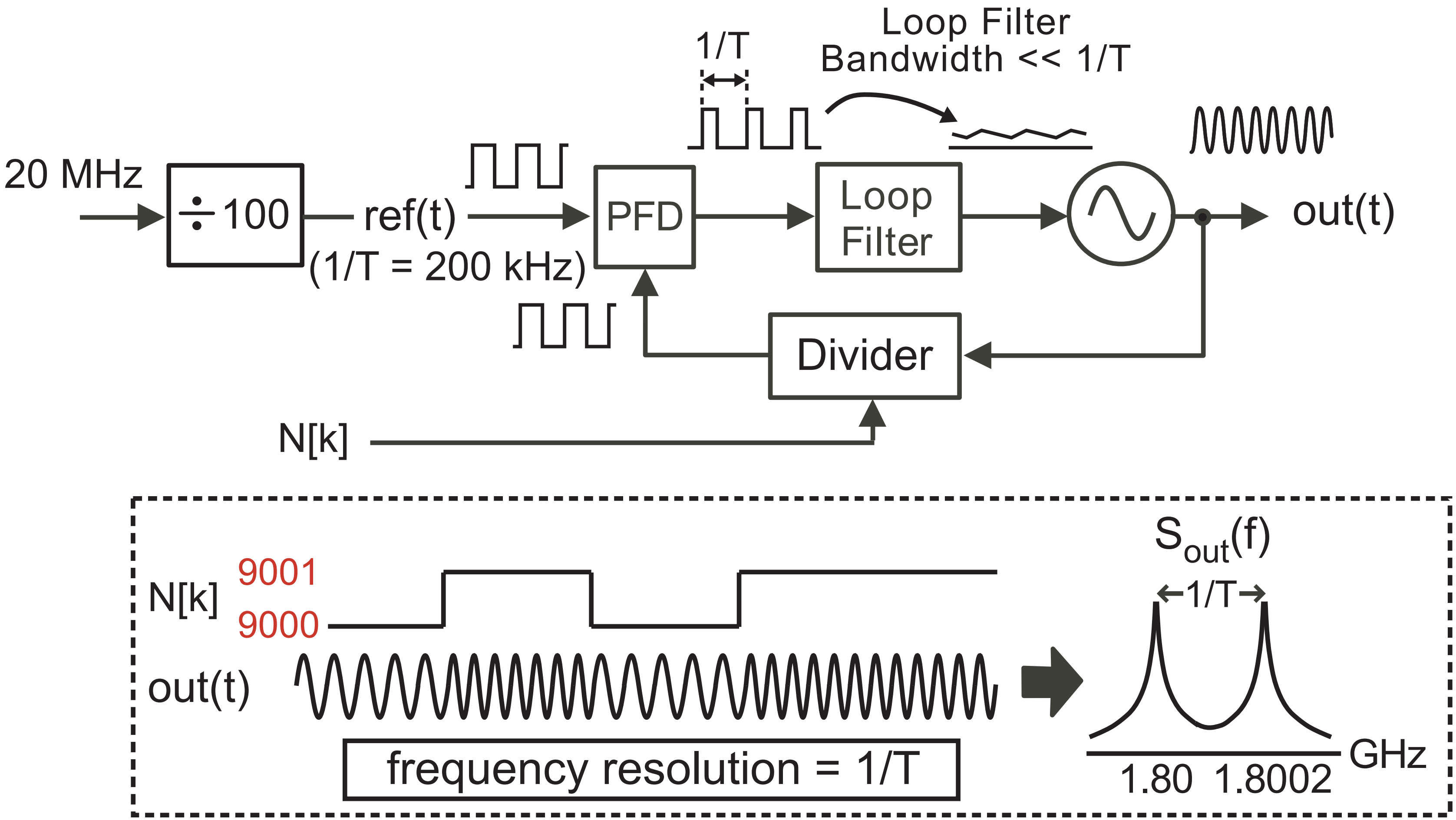
- Frequency resolution set by reference frequency (1/T)
  - Higher resolution achieved by lowering 1/T

Increasing Resolution in Integer-N Synthesizers



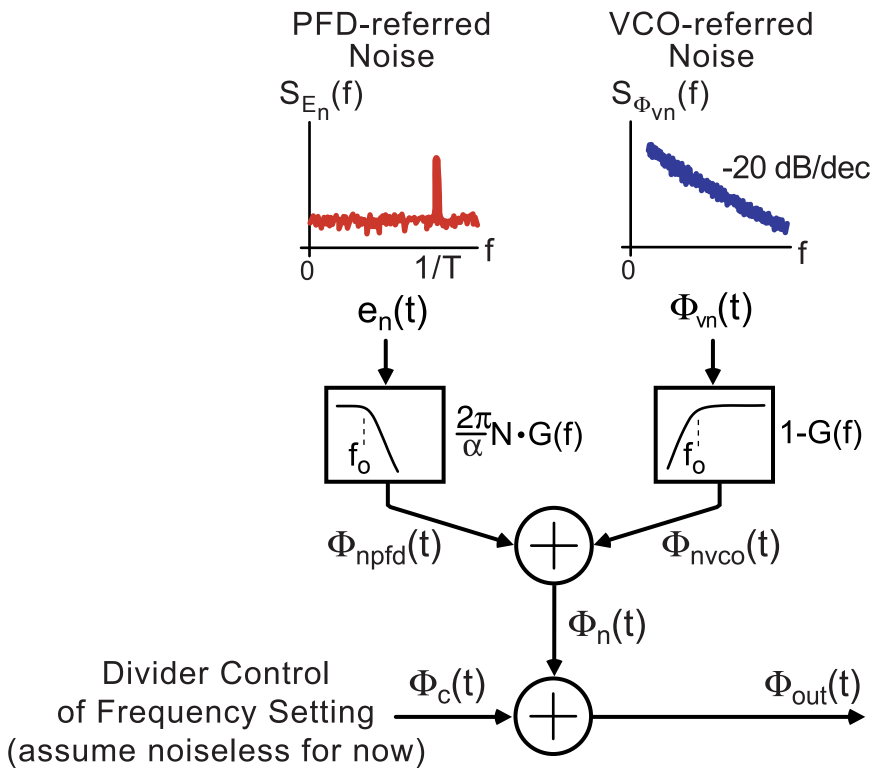
- **Use a reference divider to achieve lower 1/T**
  - **Leads to a low PLL bandwidth (< 20 kHz here)**

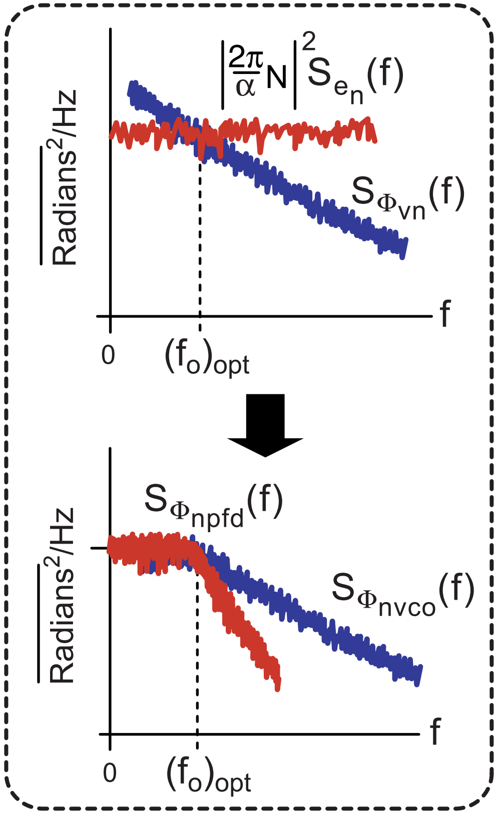
The Issue of Noise



- Lower 1/T leads to higher divide value
  - Increases PFD noise at synthesizer output

Modeling PFD Noise Multiplication

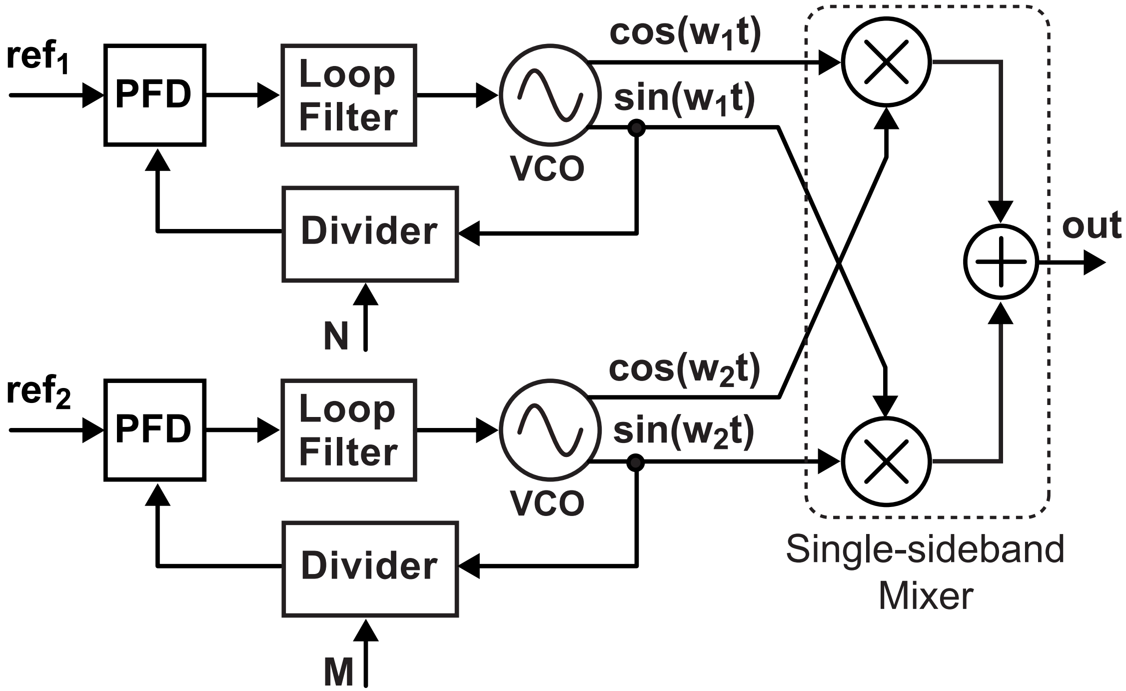




- **Influence of PFD noise seen in model from Lecture 16**
  - **PFD spectral density multiplied by N^2 before influencing PLL output phase noise**

High divide values ➡ high phase noise at low frequencies

Dual-Loop Frequency Synthesizer



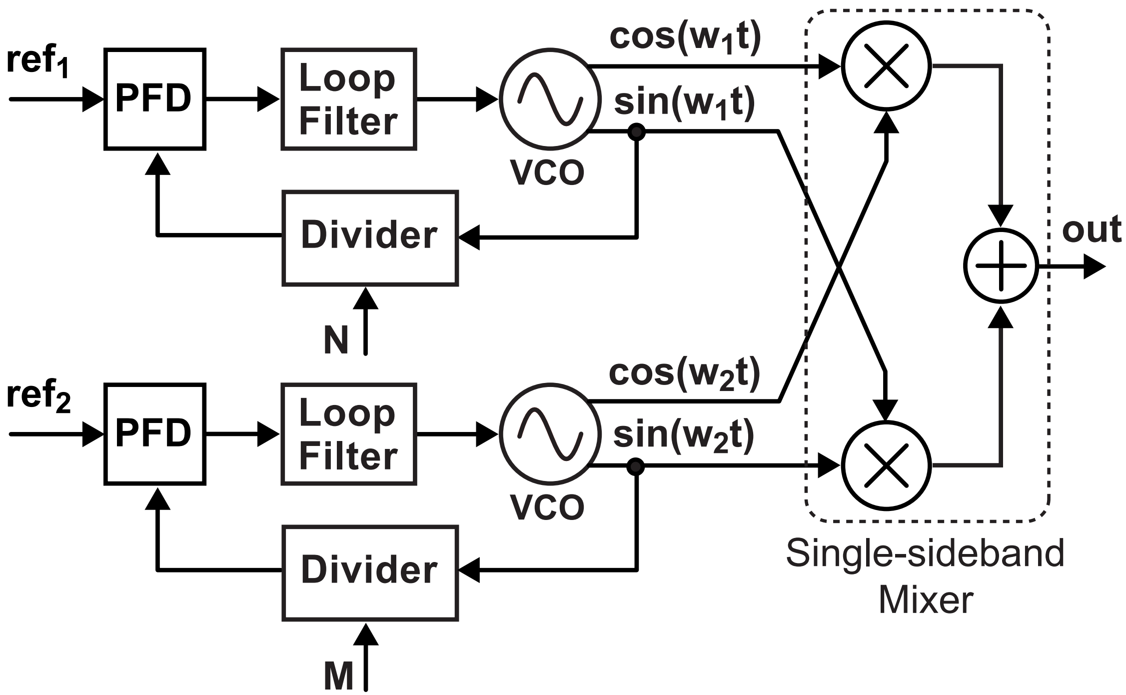
- **Overall synthesizer output**

$$out(t) = \cos(w_1 t)\cos(w_2 t) + \sin(w_1 t)\sin(w_2 t)$$

- **From trigonometry: cos(A-B) = cosAcosB+sinAsinB**

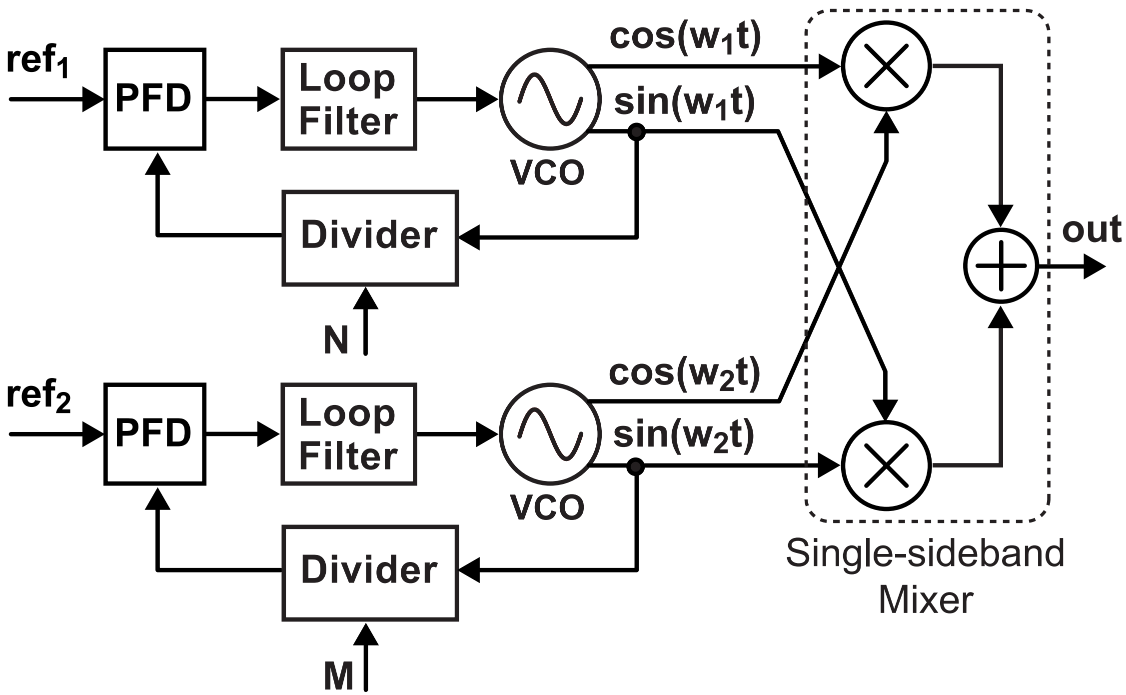
$$\Rightarrow \boxed{out(t) = \cos((w_1 - w_2)t)}$$

Advantage #1: Avoids Large Divide Values



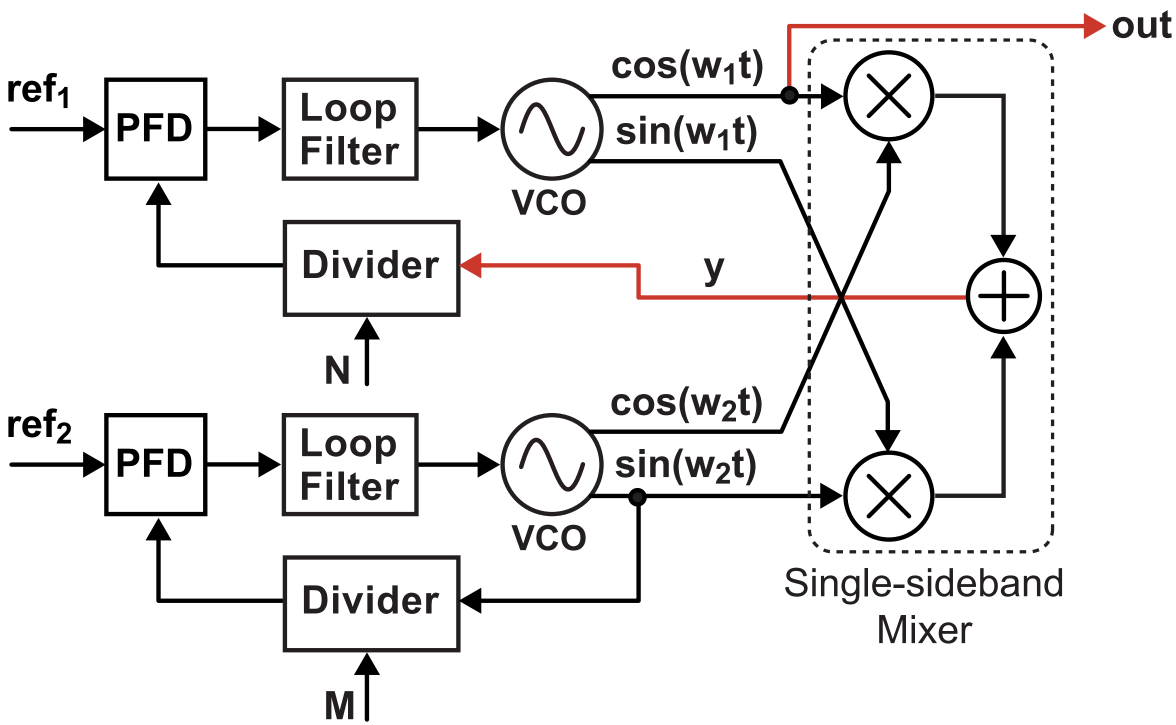
- **Choose top synthesizer to provide coarse tuning and bottom synthesizer to provide fine tuning**
  - **Choose w_1 to be high in frequency**
    - Set ref_1 to be high to avoid large N ➡ low resolution
  - **Choose w_2 to be low in frequency**
    - Allows ref_2 to be low without large M ➡ high resolution

Advantage #2: Provides Suppression of VCO Noise



- **Top VCO has much more phase noise than bottom VCO due to its much higher operating frequency**
  - **Suppress top VCO noise by choosing a high PLL bandwidth for top synthesizer**
    - High PLL bandwidth possible since ref_1 is high

Alternate Dual-Loop Architecture



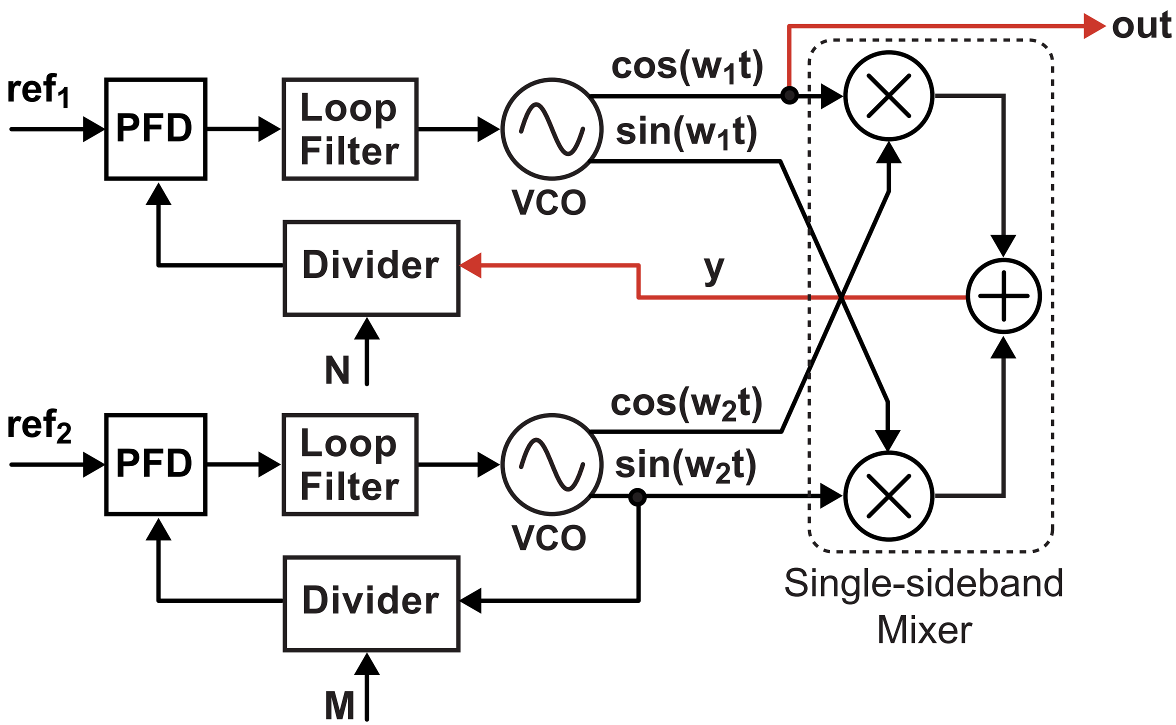
- **Calculation of output frequency**

$$y(t) = \cos((w_1 - w_2)t)$$

$$\Rightarrow \quad N w_{ref_1} = w_1 - w_2$$

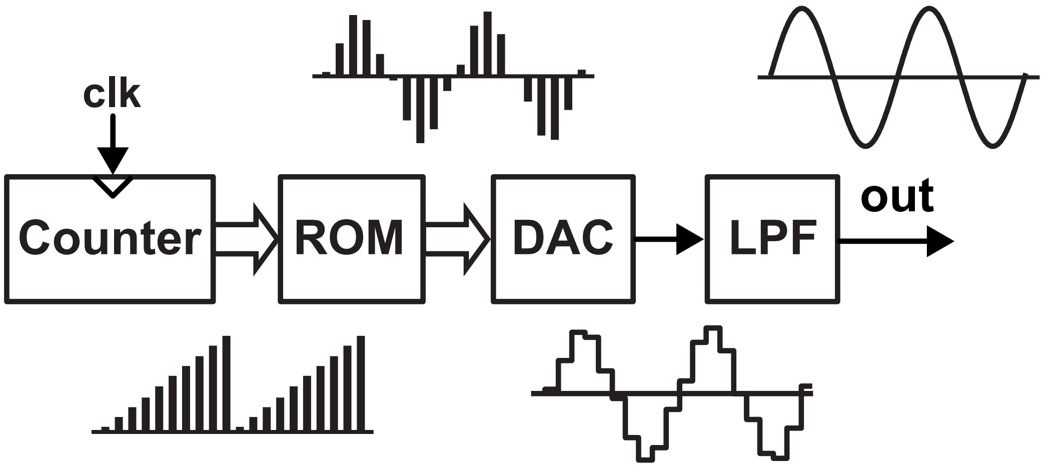
$$\Rightarrow \quad \boxed{out(t) = \cos((N w_{ref_1} + w_2)t)}$$

Advantage of Alternate Dual-Loop Architecture



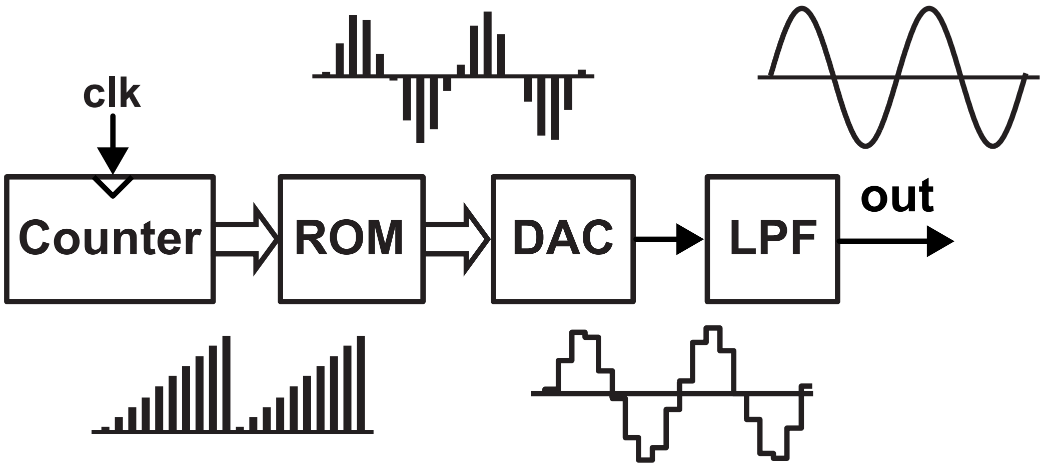
- Issue: a practical single-sideband mixer implementation will produce a spur at frequency $w_1 + w_2$
- PLL bandwidth of top synthesizer can be chosen low enough to suppress the single-sideband spur
  - Negative: lower suppression of top VCO noise

Direct Digital Synthesis (DDS)



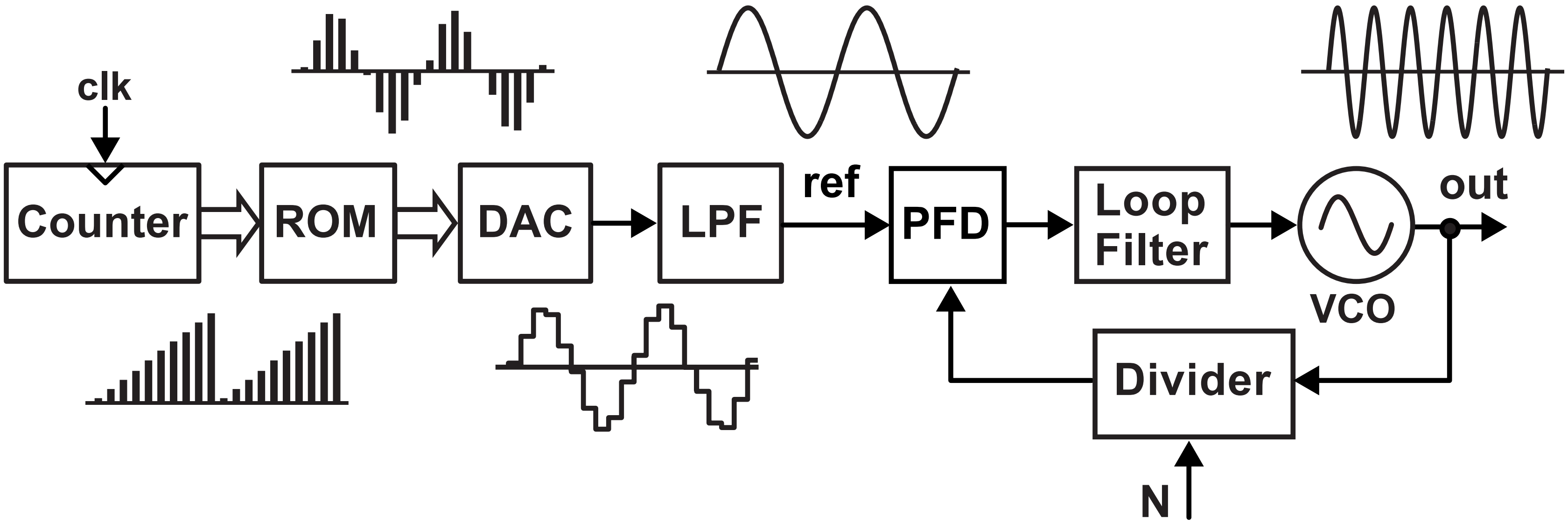
- **Encode sine-wave values in a ROM**
- **Create sine-wave output by indexing through ROM and feeding its output to a DAC and lowpass filter**
  - **Speed at which you index through ROM sets frequency of output sine-wave**
    - Speed of indexing is set by increment value on counter (which is easily adjustable in a digital manner)

Pros and Cons of Direct Digital Synthesis



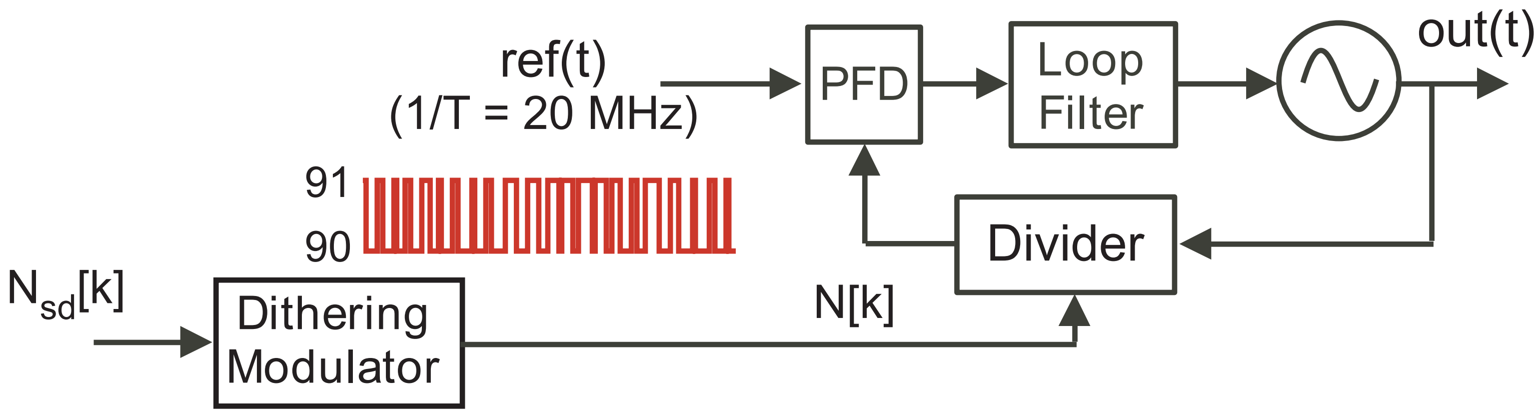
- **Advantages**
  - Very fast adjustment of frequency
  - Very high resolution can be achieved
  - Highly digital approach
- **Disadvantages**
  - Difficult to achieve high frequencies
  - Difficult to achieve low noise
  - Power hungry and complex

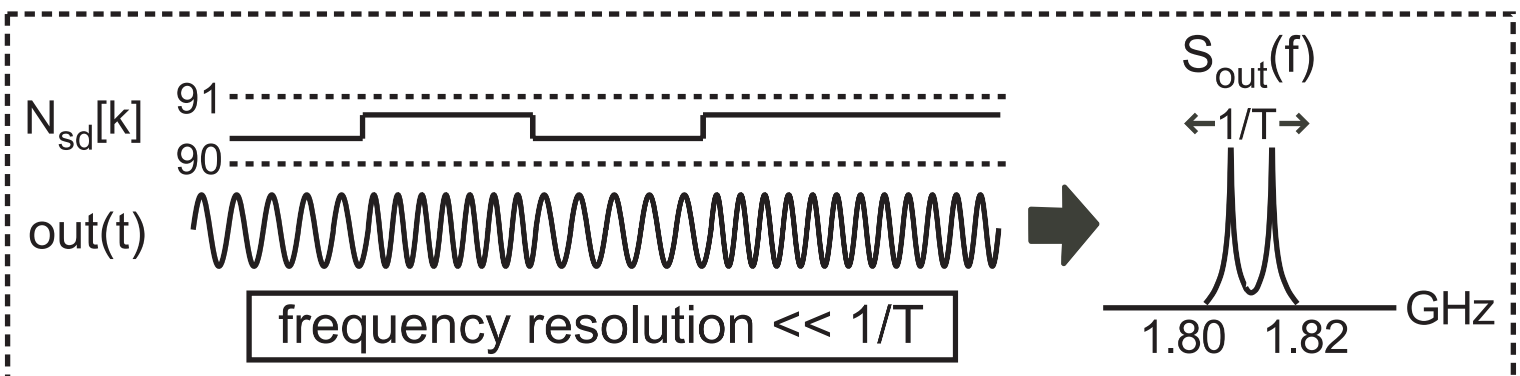
Hybrid Approach



- Use DDS to create a finely adjustable reference frequency
- Use integer-N synthesizer to multiply the DDS output frequency to much higher values
- Issues
  - Noise of DDS is multiplied by N^2
  - Complex and power hungry

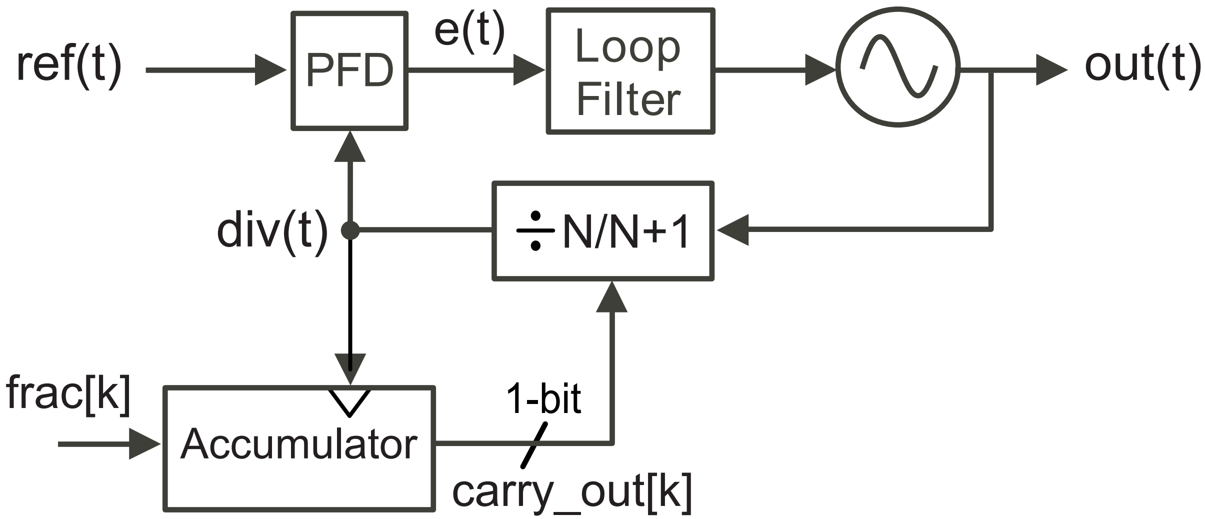
Fractional-N Frequency Synthesizers





- **Break constraint that divide value be integer**
  - **Dither divide value dynamically to achieve fractional values**
  - **Frequency resolution is now arbitrary regardless of 1/T**
- **Want high 1/T to allow a high PLL bandwidth**

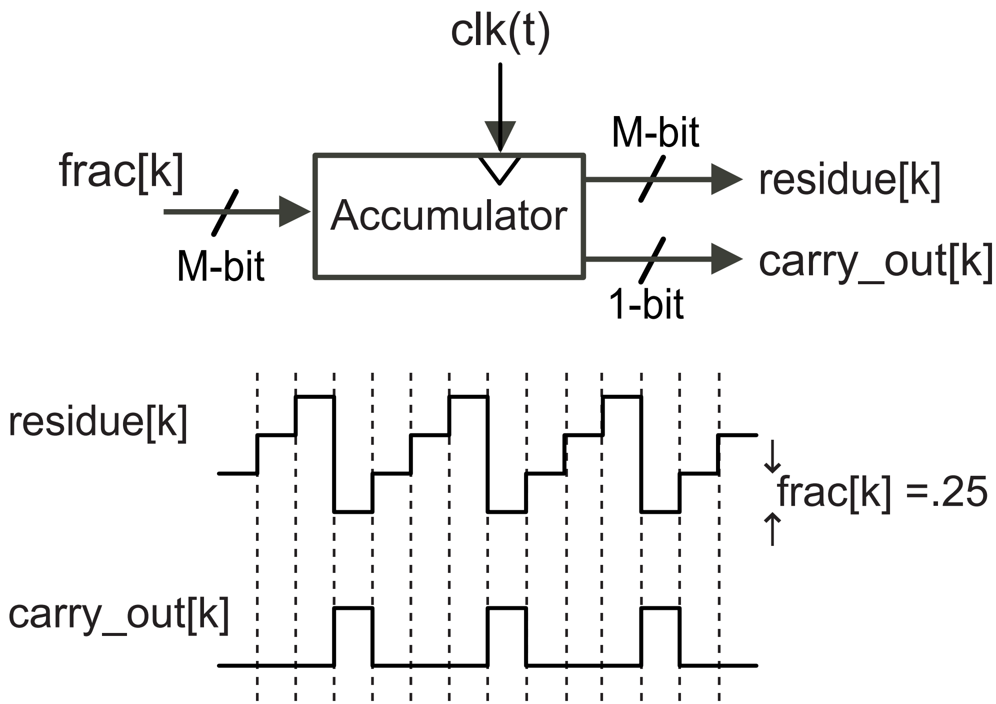
Classical Fractional-N Synthesizer Architecture



$N_{sd}[k] = N + frac[k]$

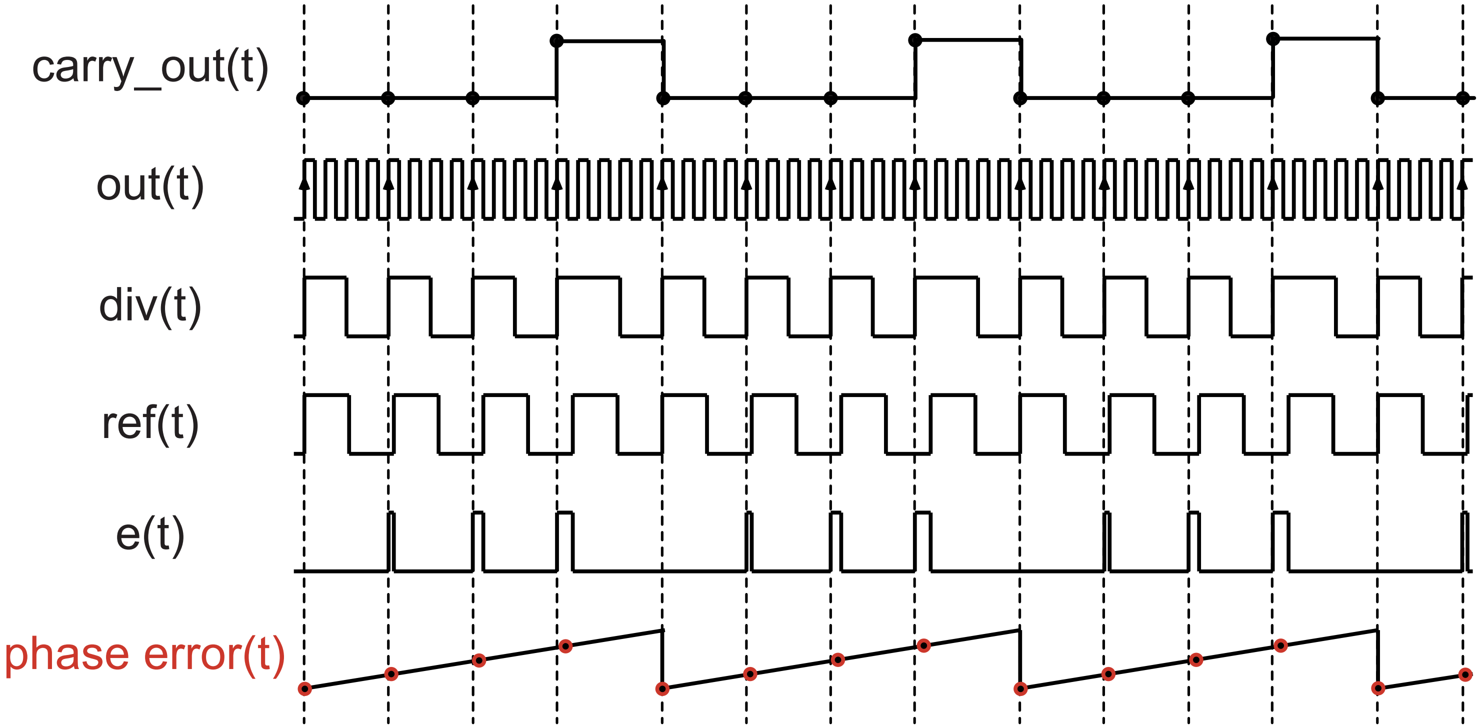
- Use an accumulator to perform dithering operation
  - Fractional input value fed into accumulator
  - Carry out bit of accumulator fed into divider

Accumulator Operation



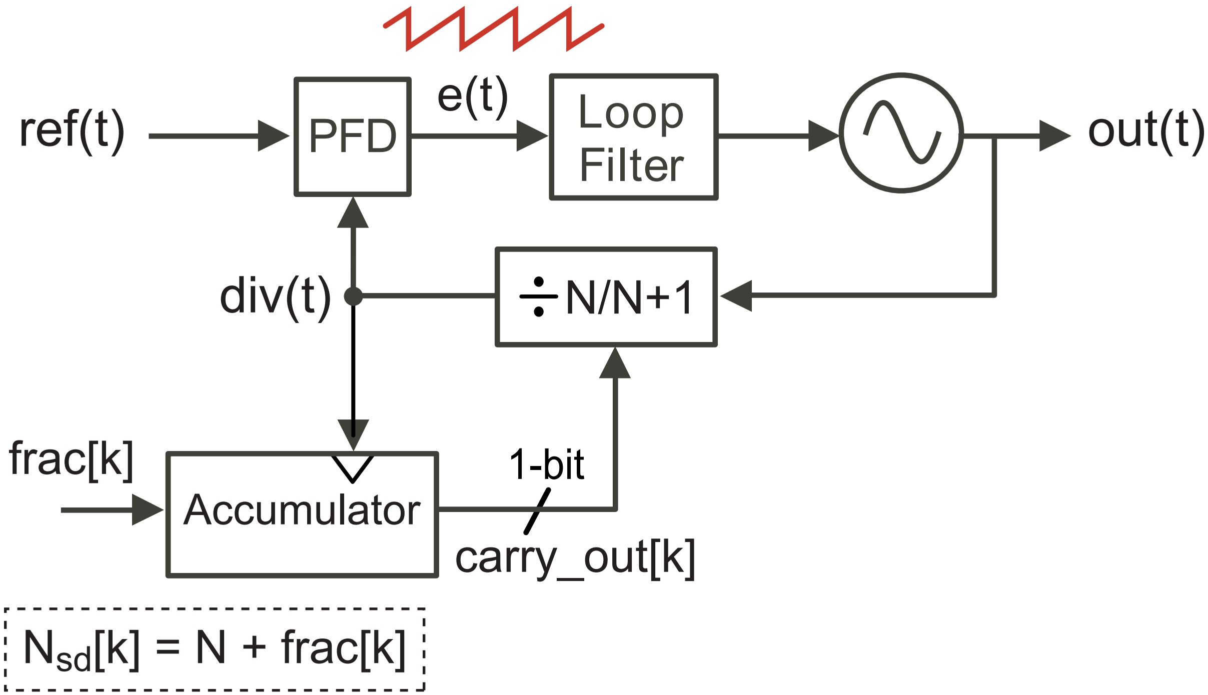
- **Carry out bit is asserted when accumulator residue reaches or surpasses its full scale value**
  - **Accumulator residue increments by input fractional value each clock cycle**

Fractional-N Synthesizer Signals with N = 4.25



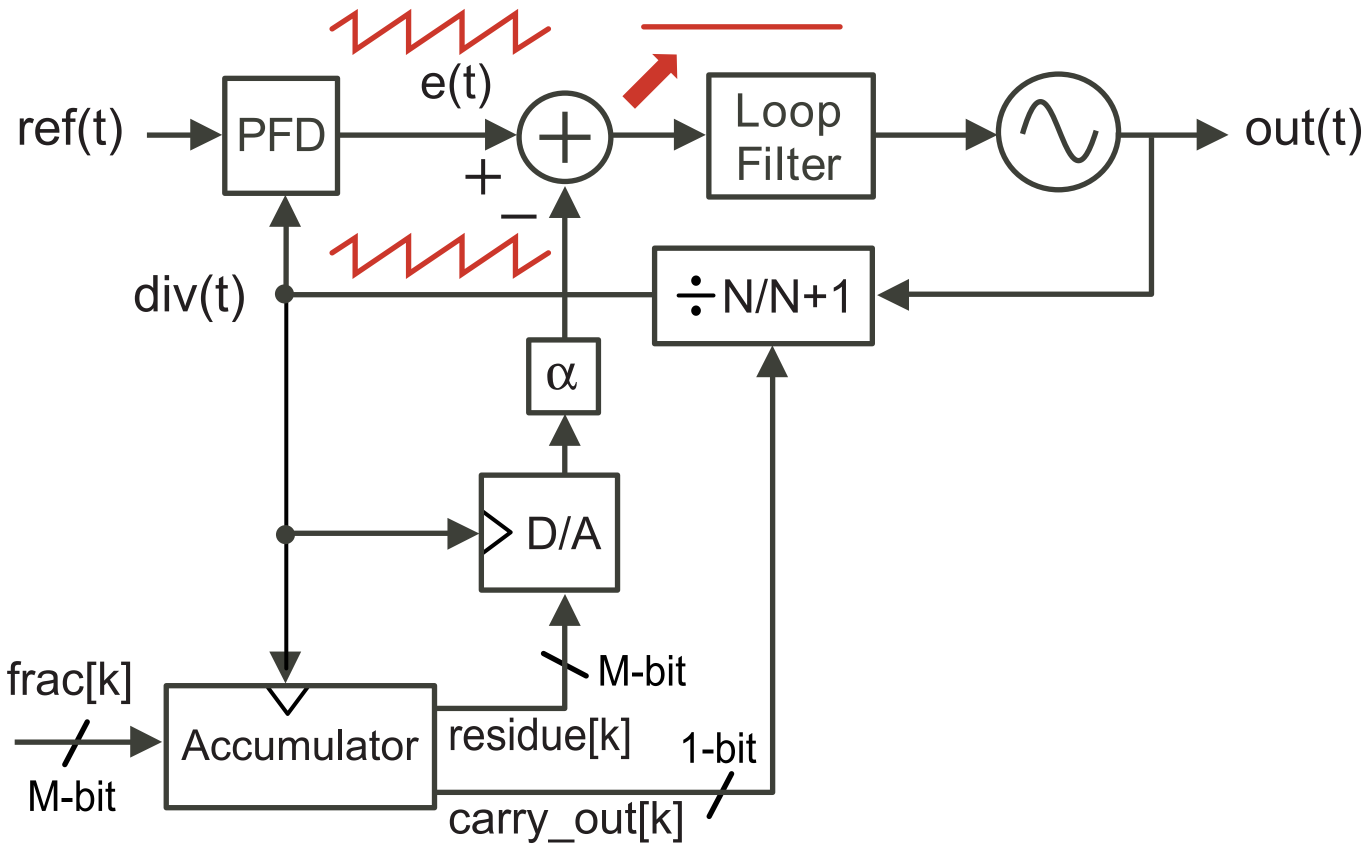
- **Divide value set at N = 4 most of the time**
  - **Resulting frequency offset causes phase error to accumulate**
  - **Reset phase error by “swallowing” a VCO cycle**
    - Achieved by dividing by 5 every 4 reference cycles

The Issue of Spurious Tones



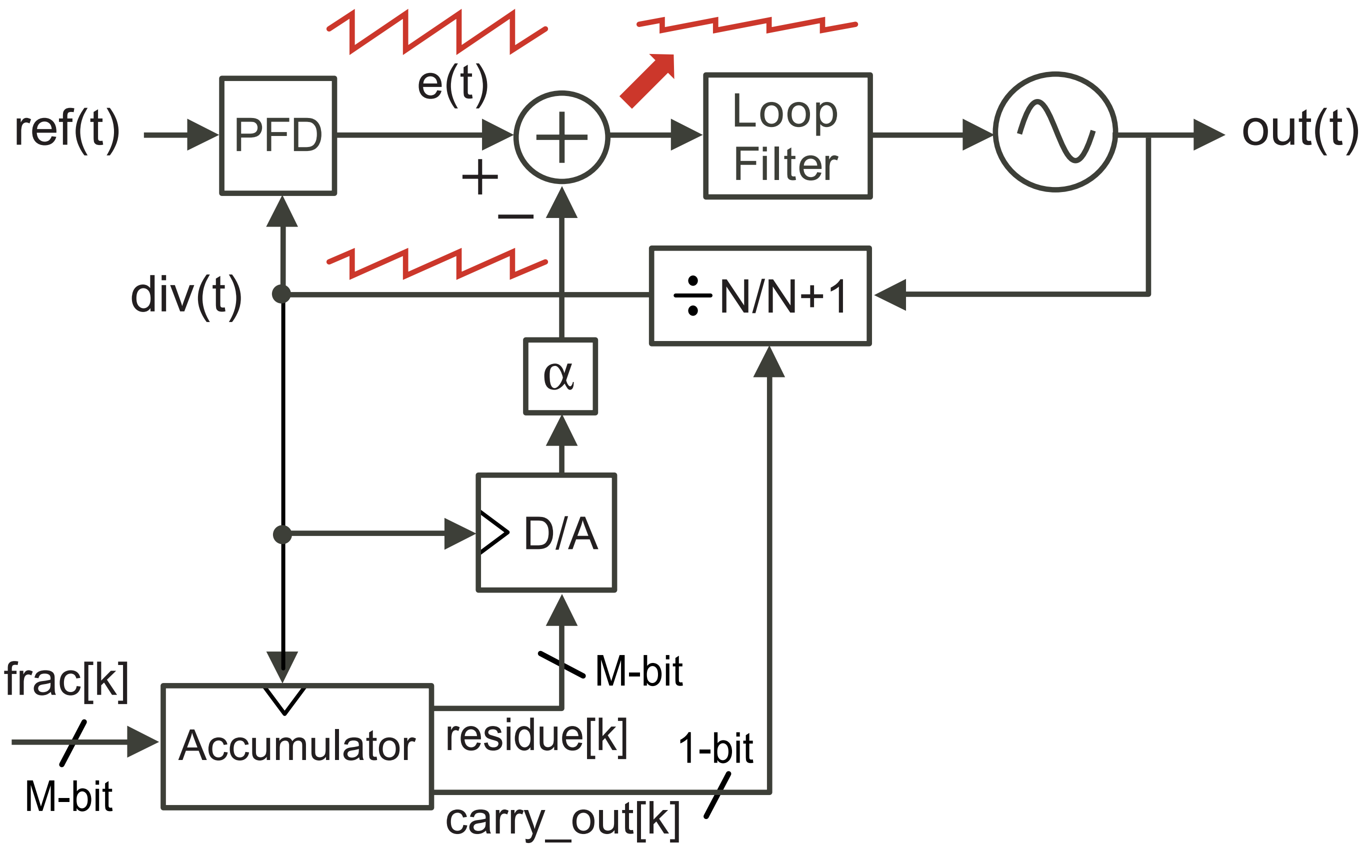
- **PFD error is periodic**
  - **Note that actual PFD waveform is series of pulses – the sawtooth waveform represents pulse width values over time**
- **Periodic error signal creates spurious tones in synthesizer output**
  - **Ruins noise performance of synthesizer**

The Phase Interpolation Technique



- **Phase error due to fractional technique is predicted by the instantaneous residue of the accumulator**
  - **Cancel out phase error based on accumulator residue**

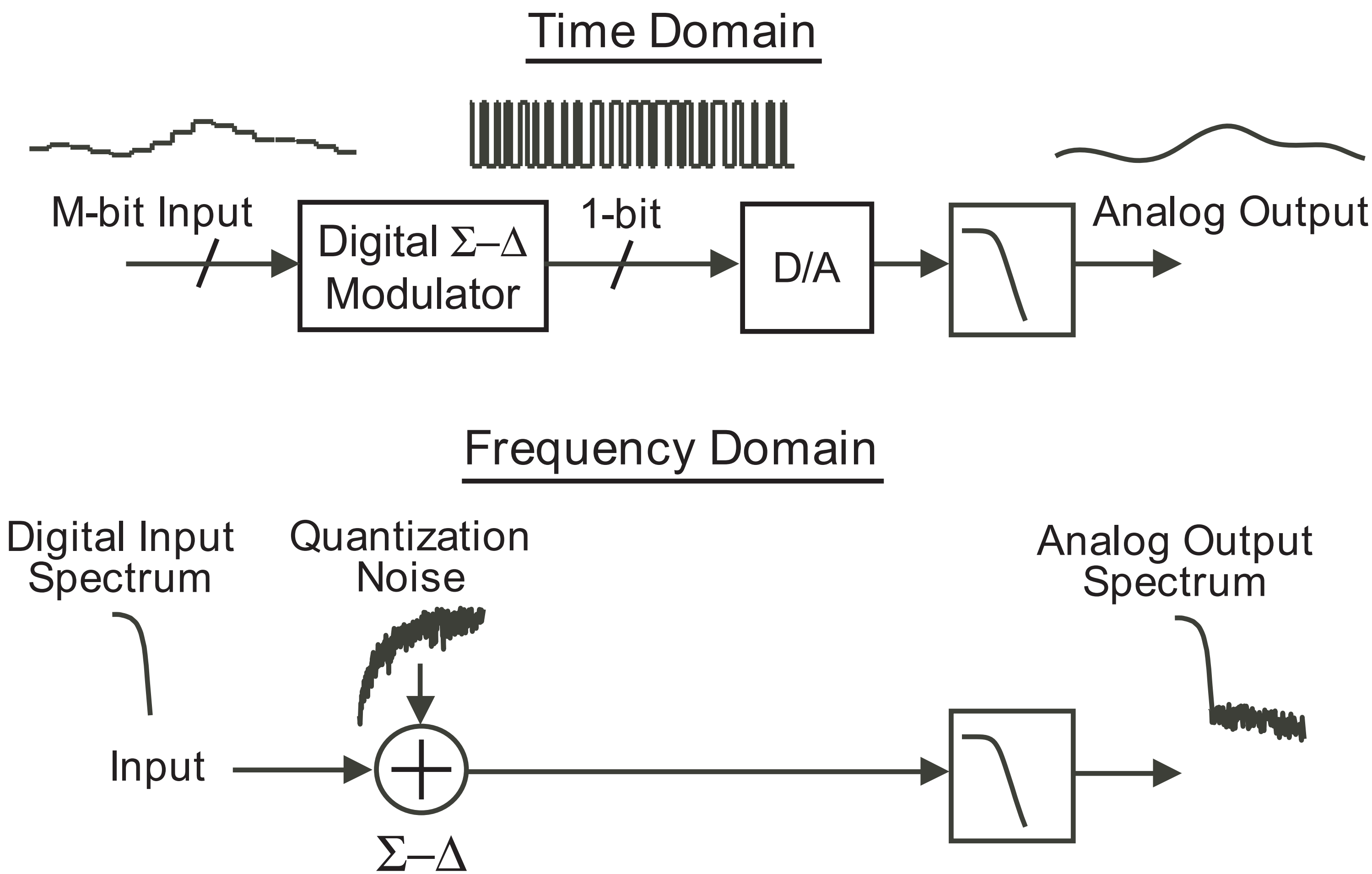
The Problem With Phase Interpolation



- **Gain matching between PFD error and scaled D/A output must be extremely precise**
  - **Any mismatch will lead to spurious tones at PLL output**

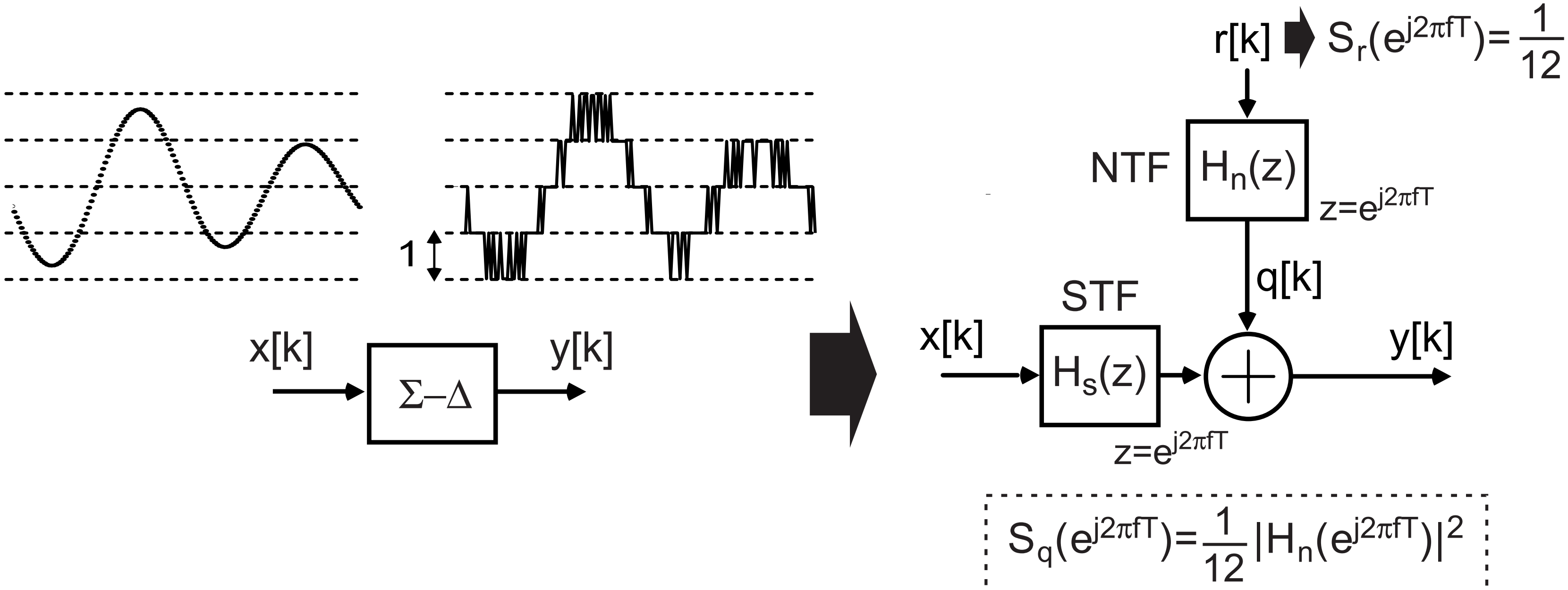
Is There a Better Way?

A Better Dithering Method: Sigma-Delta Modulation



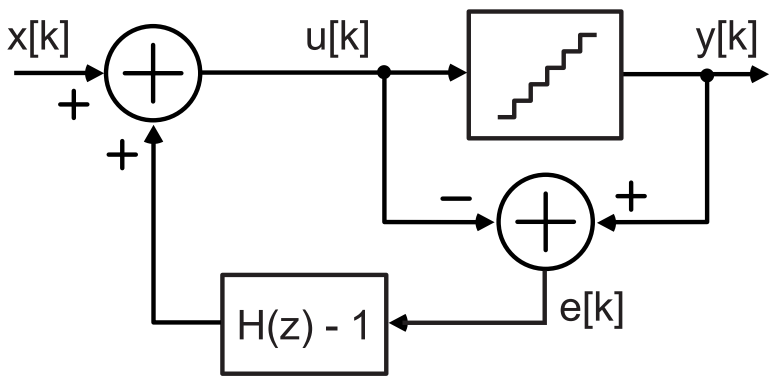
- **Sigma-Delta dithers in a manner such that resulting quantization noise is "shaped" to high frequencies**

Linearized Model of Sigma-Delta Modulator



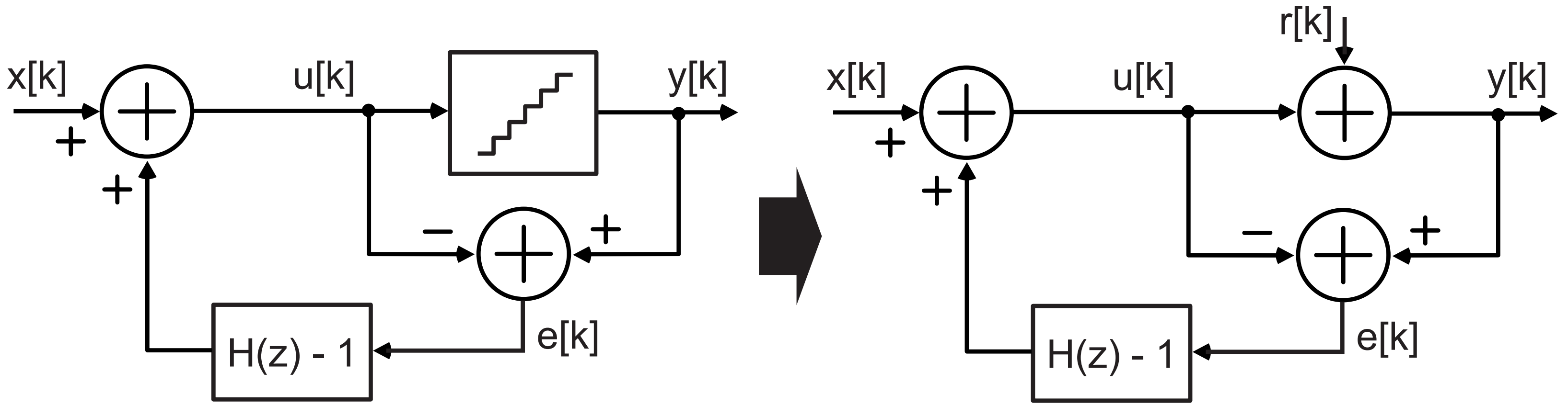
- **Composed of two transfer functions relating input and noise to output**
  - **Signal transfer function (STF)**
    - Filters input (generally undesirable)
  - **Noise transfer function (NTF)**
    - Filters (i.e., shapes) noise that is assumed to be white

Example: Cutler Sigma-Delta Topology



- **Output is quantized in a multi-level fashion**
- **Error signal, e[k], represents the quantization error**
- **Filtered version of quantization error is fed back to input**
  - **H(z) is typically a highpass filter whose first tap value is 1**
    - i.e., $H(z) = 1 + a_1 z^{-1} + a_2 z^{-2} \cdots$
  - **H(z) – 1 therefore has a first tap value of 0**
    - Feedback needs to have delay to be realizable

Linearized Model of Cutler Topology

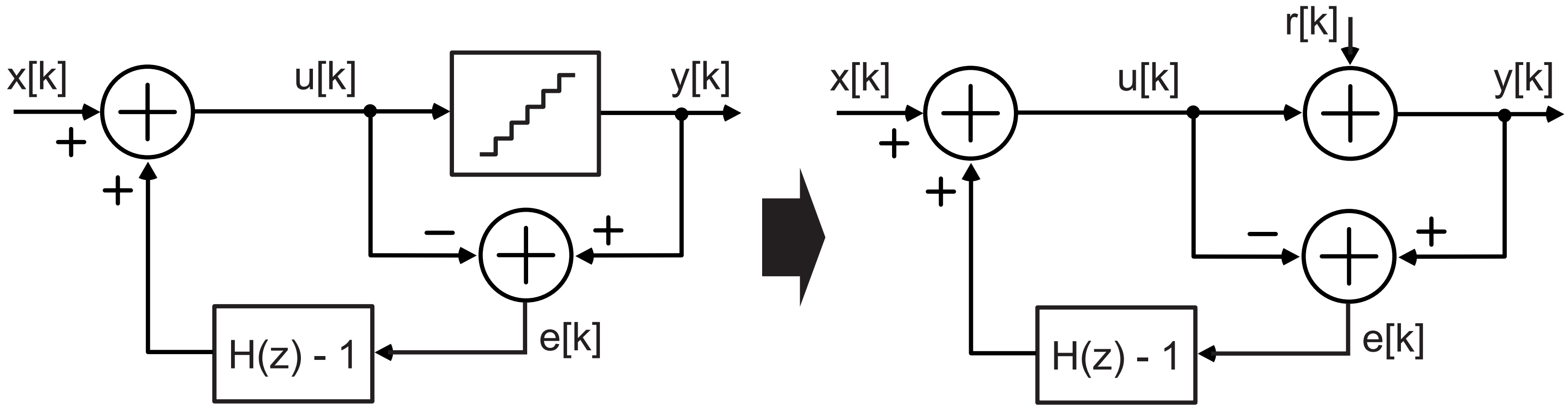


- **Represent quantizer block as a summing junction in which r[k] represents quantization error**
  - **Note:**

$$e[k] = y[k] - u[k] = (u[k] + r[k]) - u[k] = r[k]$$

- **It is assumed that r[k] has statistics similar to white noise**
  - **This is a key assumption for modeling – often not true!**

Calculation of Signal and Noise Transfer Functions



- **Calculate using Z-transform of signals in linearized model**

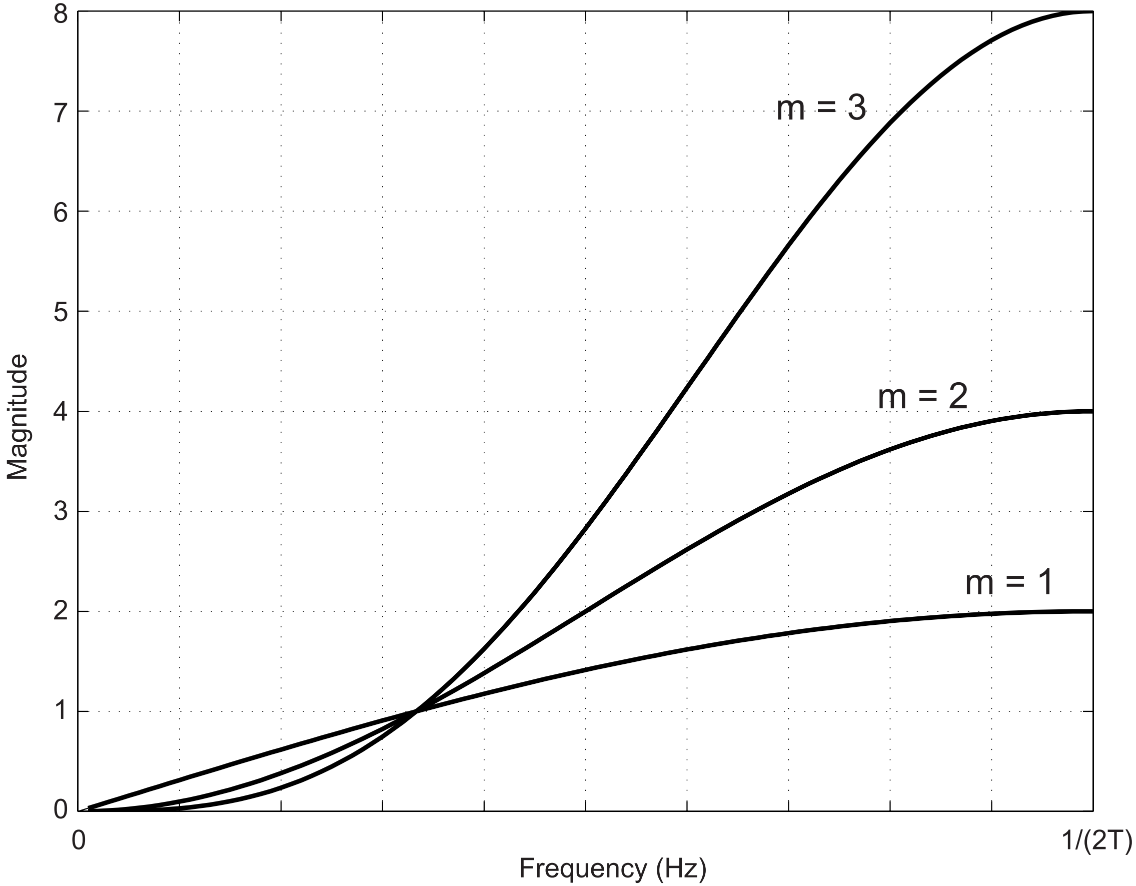
$$
\begin{aligned}
Y(z) &= U(z) + R(z) \\
&= X(z) + (H(z) - 1)E(z) + R(z) \\
&= X(z) + (H(z) - 1)R(z) + R(z) \\
&= X(z) + H(z)R(z)
\end{aligned}
$$

  - **NTF: $H_n(z) = H(z)$**
  - **STF: $H_s(z) = 1$**

A Common Choice for H(z)

$$H(z) = (1 - z^{-1})^m$$

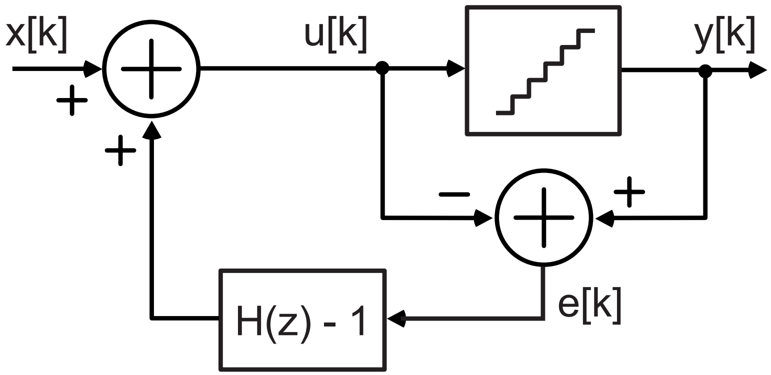
$$\Rightarrow \; |H(e^{j2\pi fT})| = |(1 - e^{-j2\pi fT})^m|$$



Example: First Order Sigma-Delta Modulator

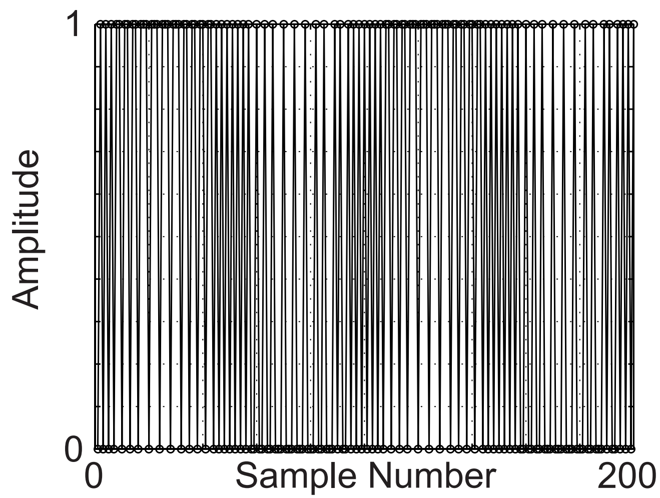
- **Choose NTF to be**

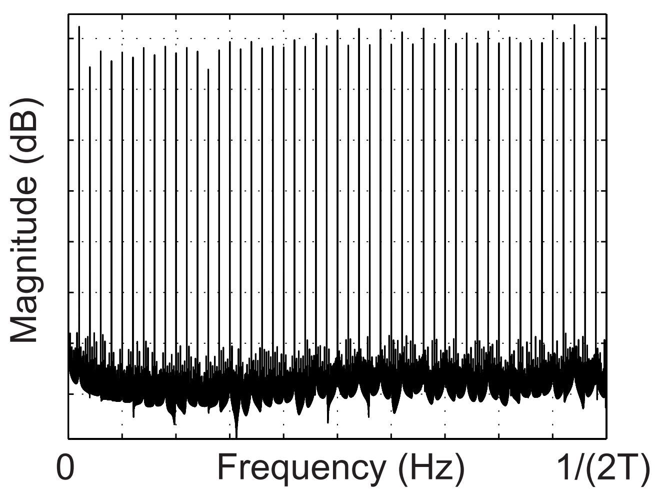
$$H_n(z) = H(z) = 1 - z^{-1}$$



- **Plot of output in time and frequency domains with input of**

$$x[k] = 0.5 + 0.25 \sin\left(\frac{2\pi}{100}k\right)$$

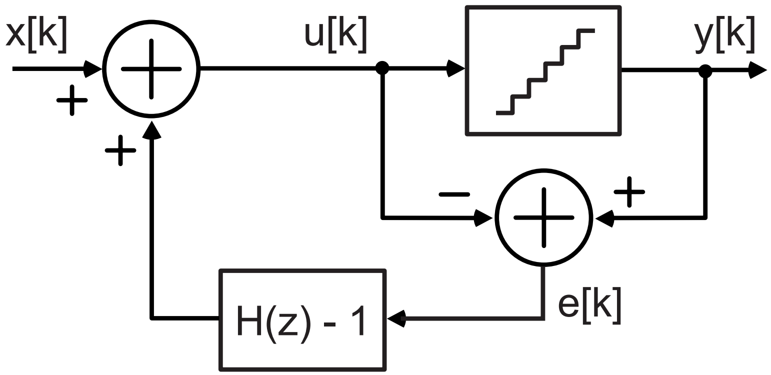




Example: Second Order Sigma-Delta Modulator

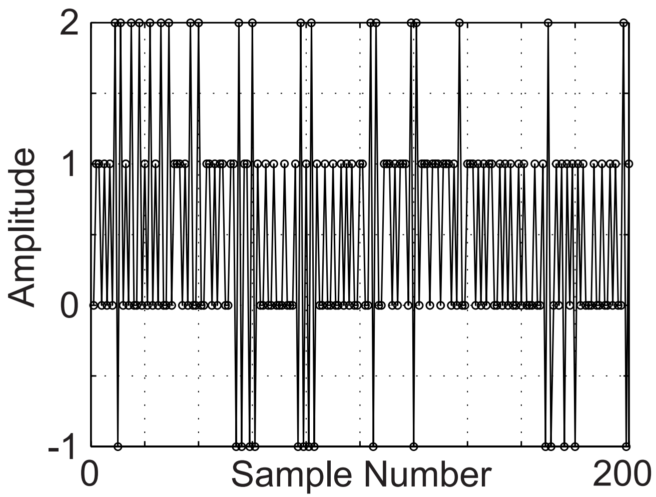
- **Choose NTF to be**

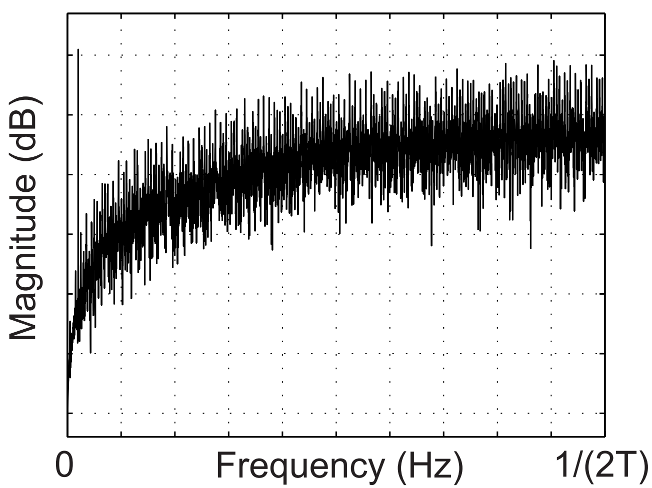
$$H_n(z) = H(z) = (1 - z^{-1})^2$$



- **Plot of output in time and frequency domains with input of**

$$x[k] = 0.5 + 0.25\sin\left(\frac{2\pi}{100}k\right)$$

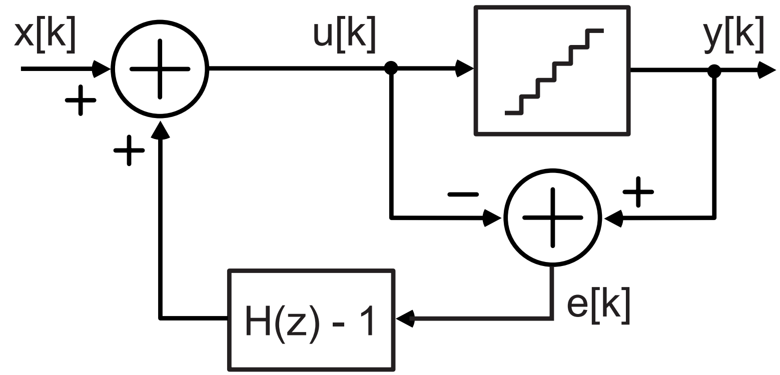




Example: Third Order Sigma-Delta Modulator

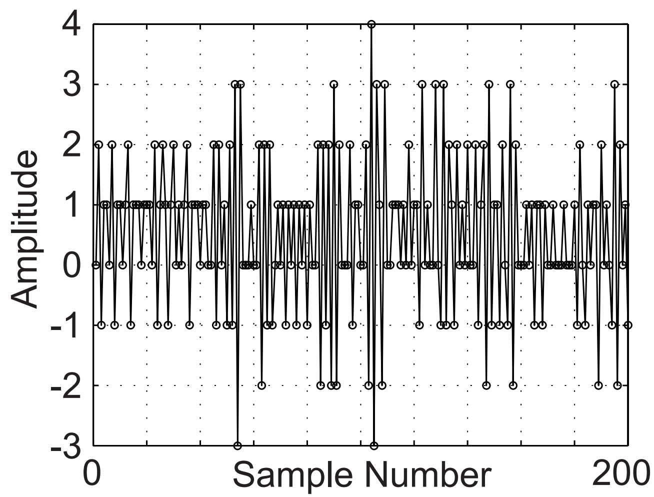
- **Choose NTF to be**

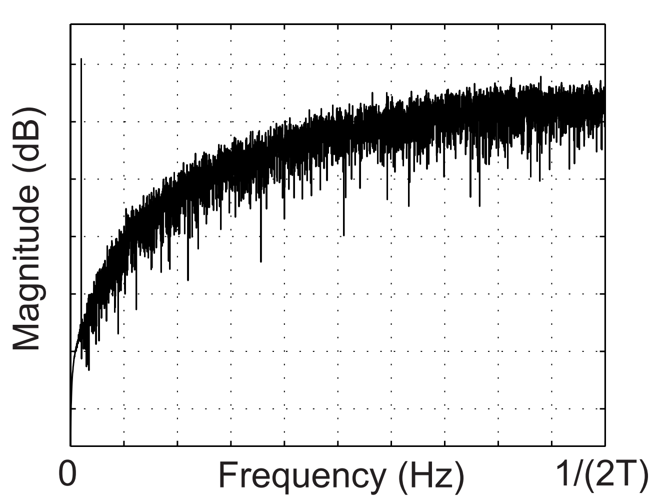
$$H_n(z) = H(z) = (1 - z^{-1})^3$$



- **Plot of output in time and frequency domains with input of**

$$x[k] = 0.5 + 0.25 \sin\left(\frac{2\pi}{100}k\right)$$





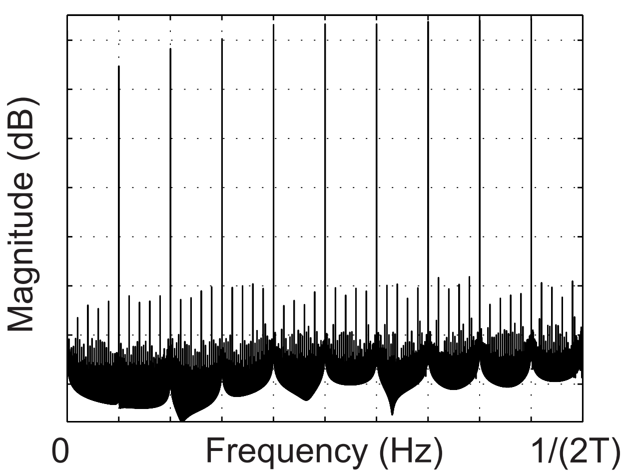
Observations

- **Low order Sigma-Delta modulators do not appear to produce “shaped” noise very well**
  - **Reason: low order feedback does not properly “scramble” relationship between input and quantization noise**
    - Quantization noise, r[k], fails to be white
- **Higher order Sigma-Delta modulators provide much better noise shaping with fewer spurs**
  - **Reason: higher order feedback filter provides a much more complex interaction between input and quantization noise**

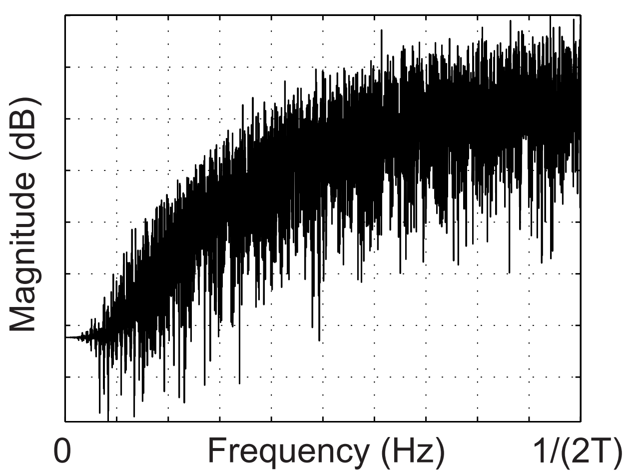
Warning: Higher Order Modulators May Still Have Tones

- **Quantization noise, r[k], is best whitened when a "sufficiently exciting" input is applied to the modulator**
  - **Varying input and high order helps to "scramble" interaction between input and quantization noise**
- **Worst input for tone generation are DC signals that are rational with a low valued denominator**
  - **Examples (third order modulator):**

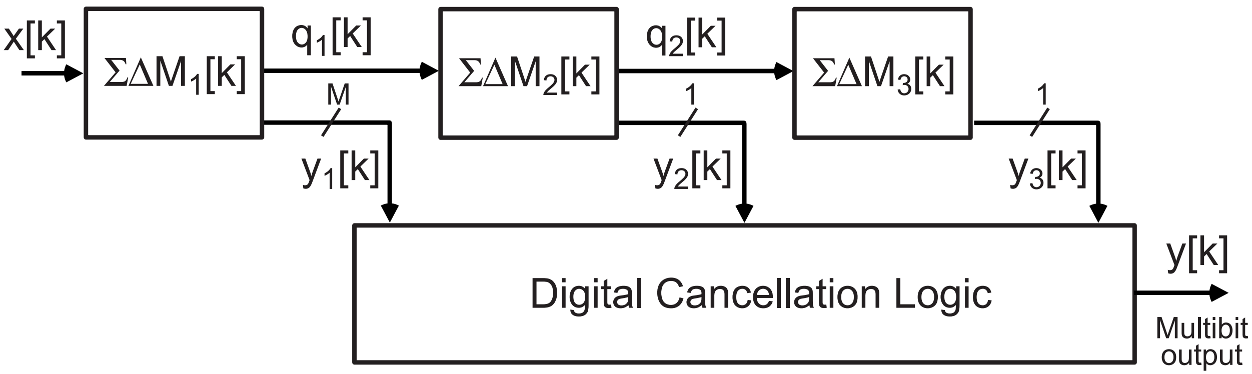
$x[k] = 0.1$



$x[k] = 0.1 + 1/1024$

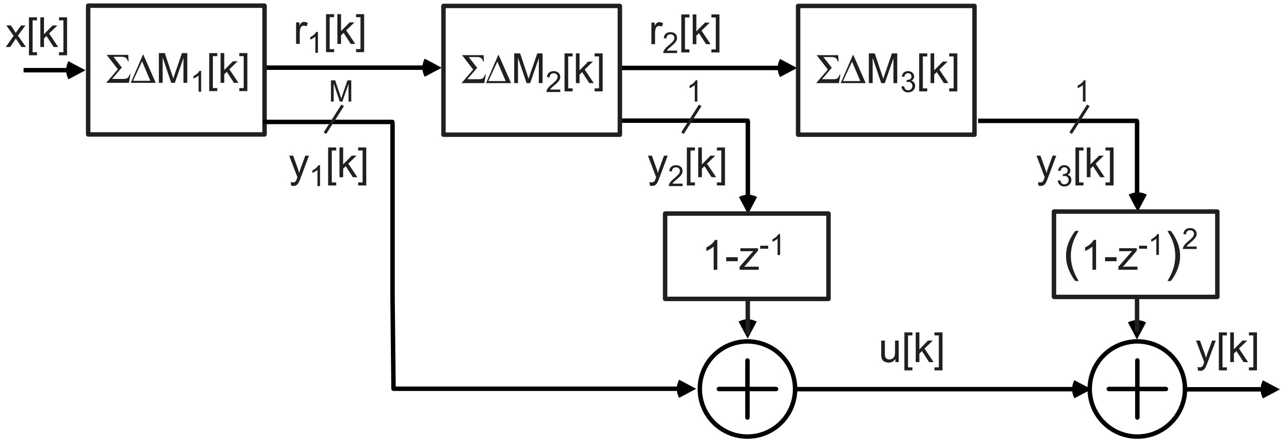


Cascaded Sigma-Delta Modulator Topologies



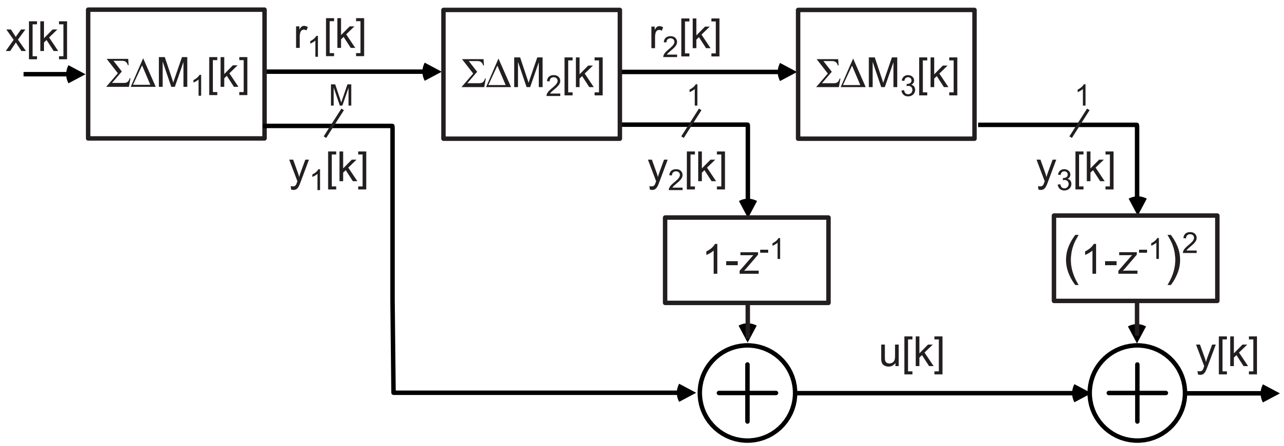
- **Achieve higher order shaping by cascading low order sections and properly combining their outputs**
- **Advantage over single loop approach**
  - **Allows pipelining to be applied to implementation**
    - High speed or low power applications benefit
- **Disadvantages**
  - **Relies on precise matching requirements when combining outputs (not a problem for digital implementations)**
  - **Requires multi-bit quantizer (single loop does not)**

MASH topology



- **Cascade first order sections**
- **Combine their outputs after they have passed through digital differentiators**

Calculation of STF and NTF for MASH topology (Step 1)



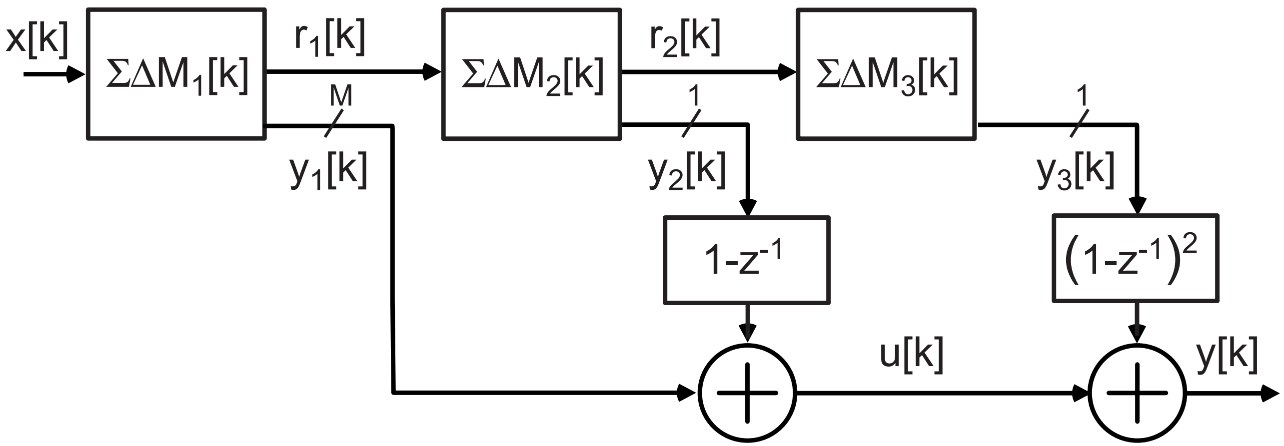
- **Individual output signals of each first order modulator**

$$\begin{aligned} y_1(z) &= x(z) - (1-z^{-1})r_1(z) \\ y_2(z) &= r_1(z) - (1-z^{-1})r_2(z) \\ y_3(z) &= r_2(z) - (1-z^{-1})r_3(z) \end{aligned}$$

- **Addition of filtered outputs**

$$\begin{array}{rr} & y_1(z) \\ + & (1-z^{-1})y_2(z) \\ + & (1-z^{-1})^2 y_2(z) \\ \hline = & x(z) - (1-z^{-1})^3 r_3(z) \end{array}$$

Calculation of STF and NTF for MASH topology (Step 1)

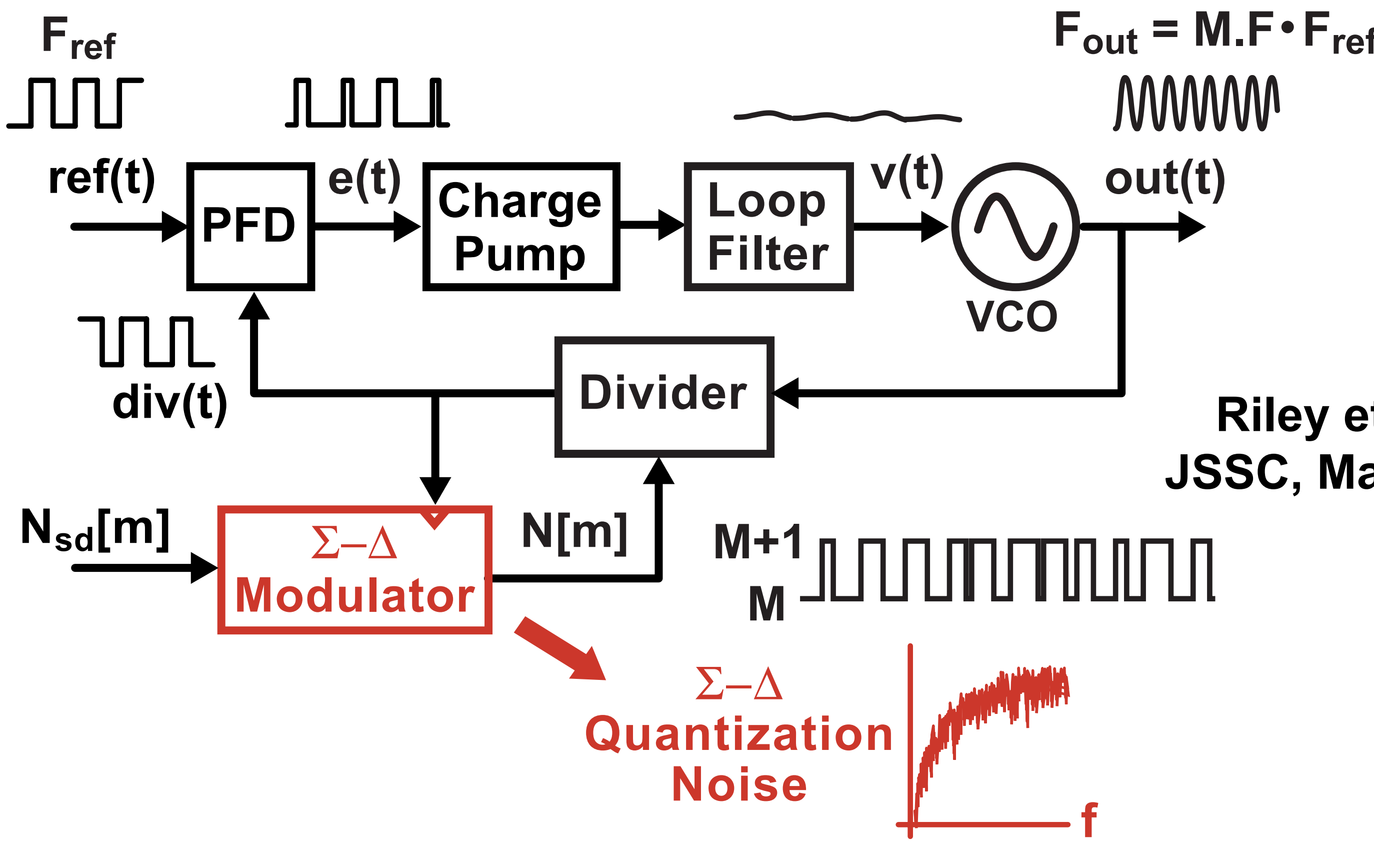


- **Overall modulator behavior**

$$y(z) = x(z) - (1 - z^{-1})^3 r_3(z)$$

  - **STF: $H_s(z) = 1$**
  - **NTF: $H_n(z) = (1 - z^{-1})^3$**

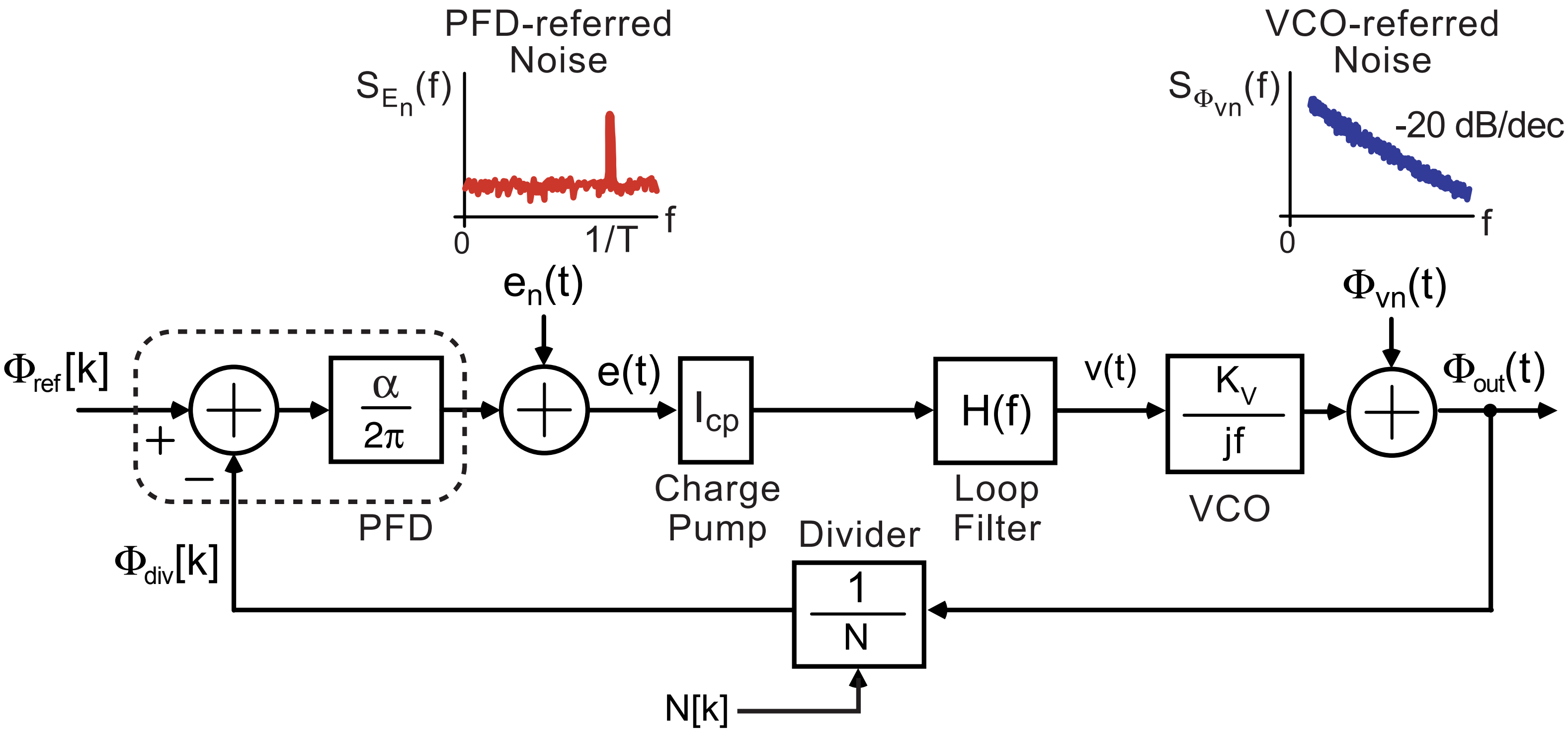
Sigma-Delta Frequency Synthesizers



Riley et. al., JSSC, May 1993

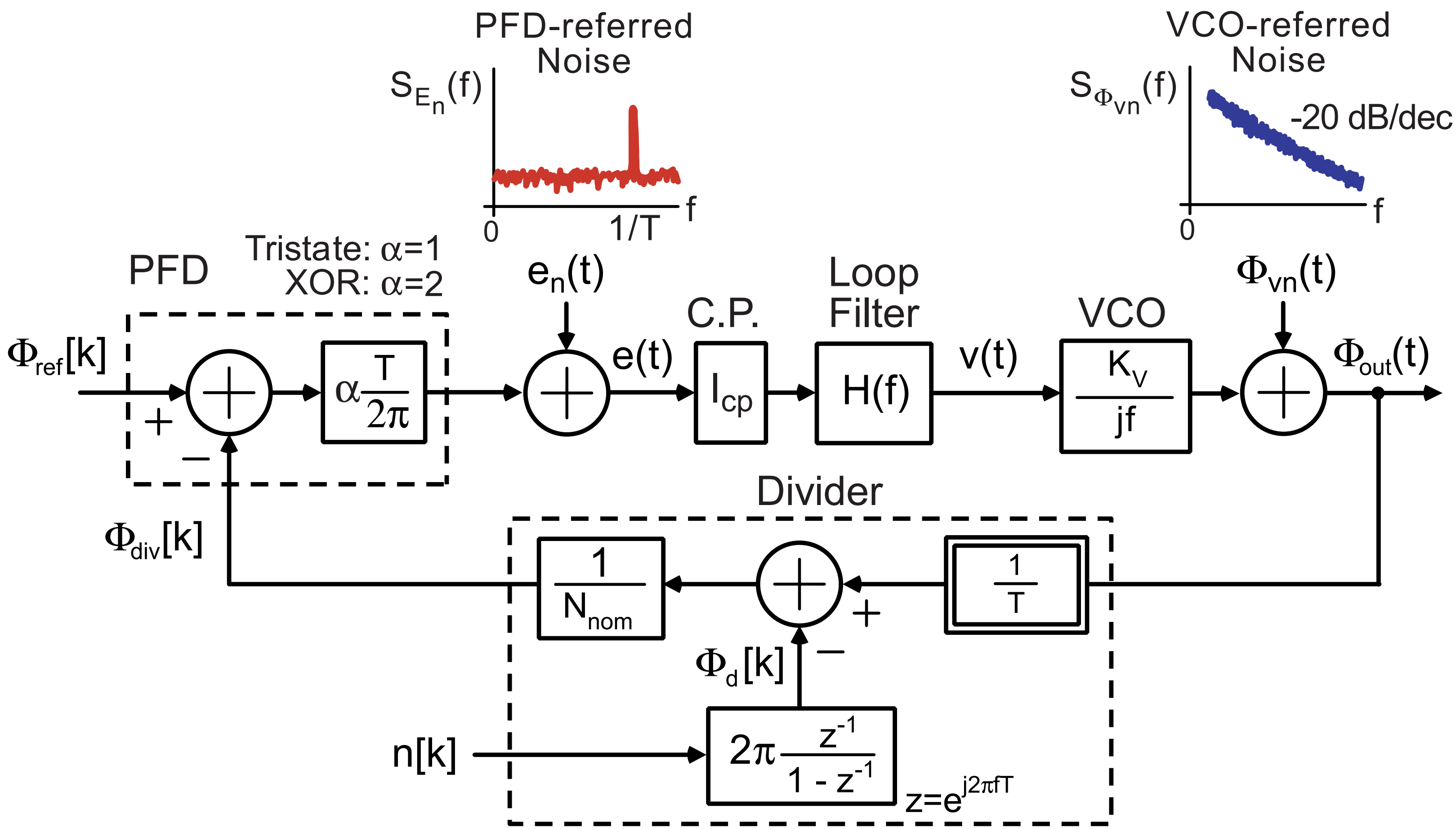
- Use Sigma-Delta modulator rather than accumulator to perform dithering operation
  - Achieves much better spurious performance than classical fractional-N approach

Background: The Need for A Better PLL Model



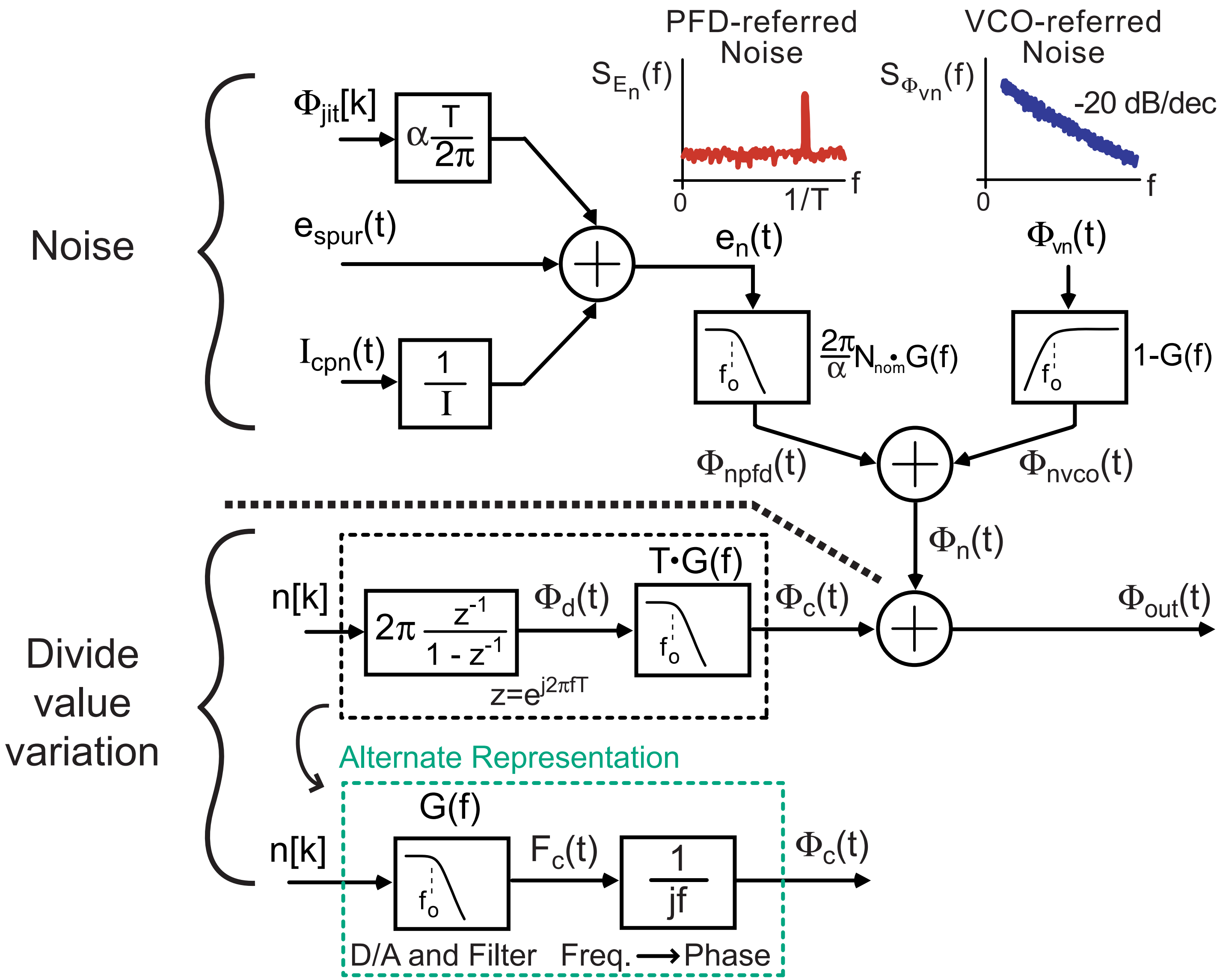
- **Classical PLL model**
  - **Predicts impact of PFD and VCO referred noise sources**
  - **Does not allow straightforward modeling of impact due to divide value variations**
    - This is a problem when using fractional-N approach

A PLL Model Accommodating Divide Value Variations

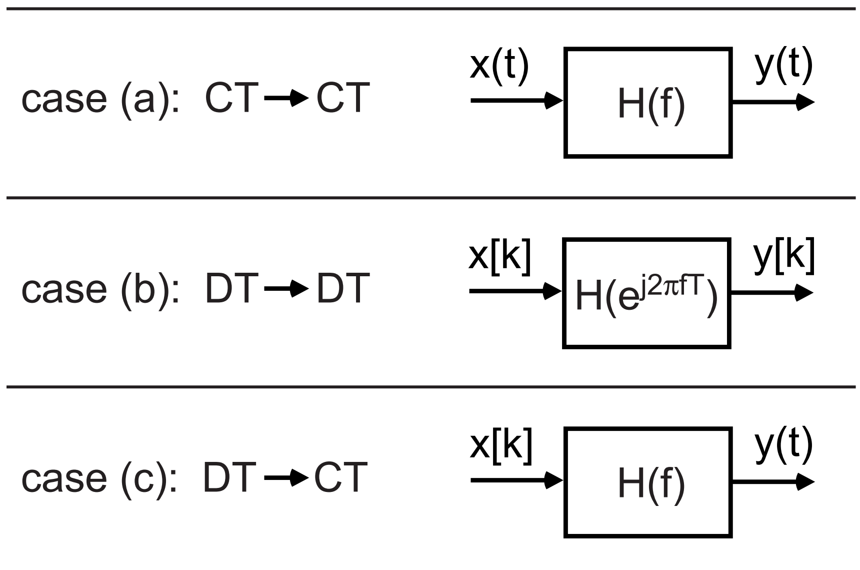


- **See derivation in Perrott et. al., "A Modeling Approach for Sigma-Delta Fractional-N Frequency Synthesizers …", JSSC, Aug 2002**

Parameterized Version of New Model

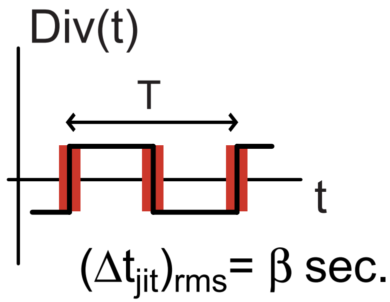


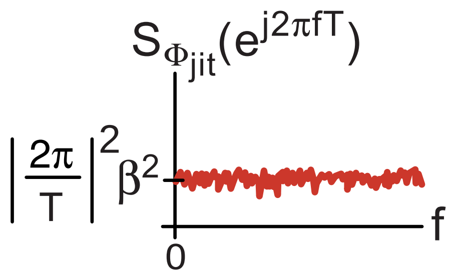
Spectral Density Calculations

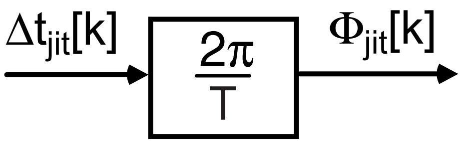


- **Case (a):** $$S_y(f) = |H(f)|^2 S_x(f)$$
- **Case (b):** $$S_y(e^{j2\pi fT}) = |H(e^{j2\pi fT})|^2 S_x(e^{j2\pi fT})$$
- **Case (c):** $$S_y(f) = \frac{1}{T}|H(f)|^2 S_x(e^{j2\pi fT})$$

Example: Calculate Impact of Ref/Divider Jitter (Step 1)



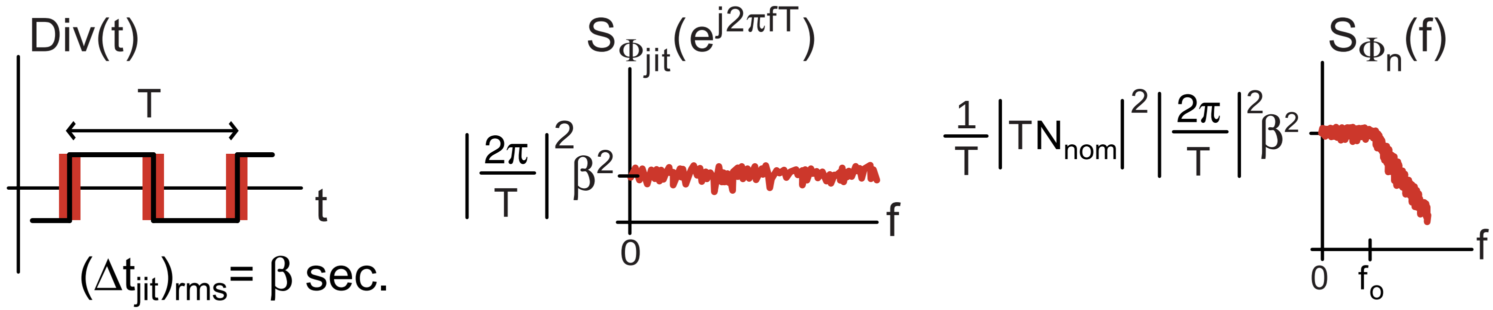




- **Assume jitter is white**
  - **i.e., each jitter value independent of values at other time instants**
- **Calculate spectra for discrete-time input and output**
  - **Apply case (b) calculation**

$$S_{\Delta t_{jit}}(e^{j2\pi fT}) = \beta^2 \;\; \Rightarrow \;\; S_{\Phi_{jit}}(e^{j2\pi fT}) = \left|\frac{2\pi}{T}\right|^2 \beta^2$$

Example: Calculate Impact of Ref/Divider Jitter (Step 2)



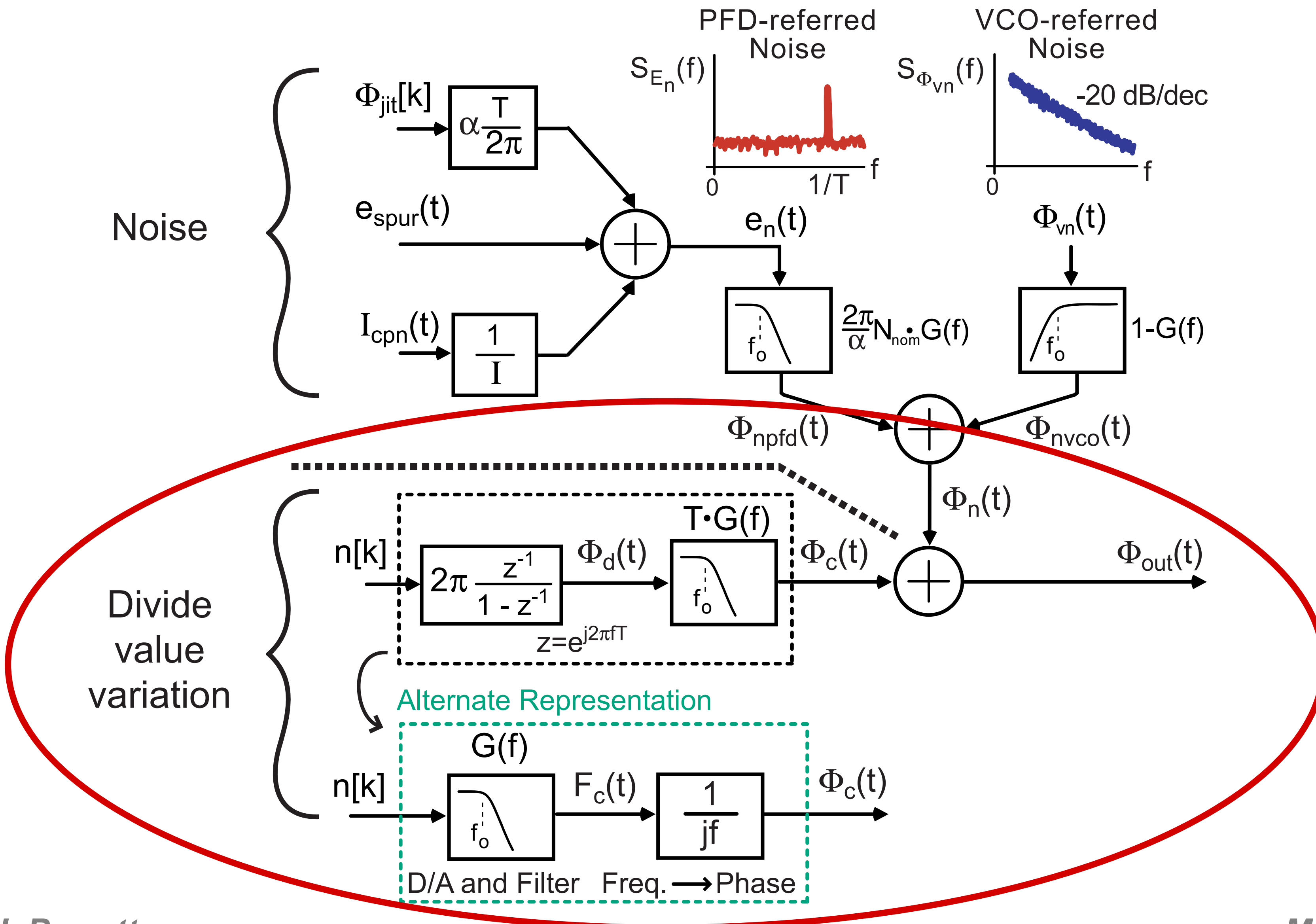
- **Compute impact on output phase noise of synthesizer**
  - **We now apply case (c) calculation**

$$S_{\Phi_n}(f) = \frac{1}{T}|TN_{nom}G(f)|^2 S_{\Phi_{jit}}(e^{j2\pi fT})$$

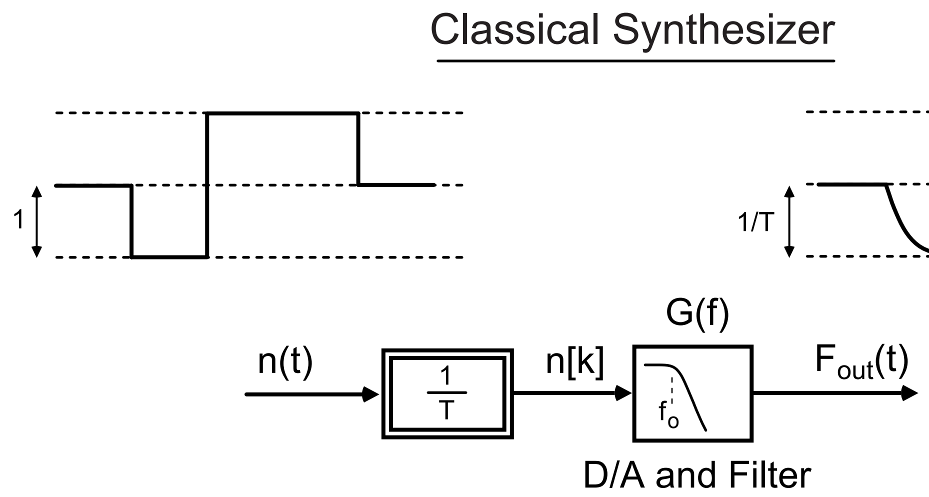
$$= \frac{1}{T}|TN_{nom}G(f)|^2 \left|\frac{2\pi}{T}\right|^2 \beta^2$$

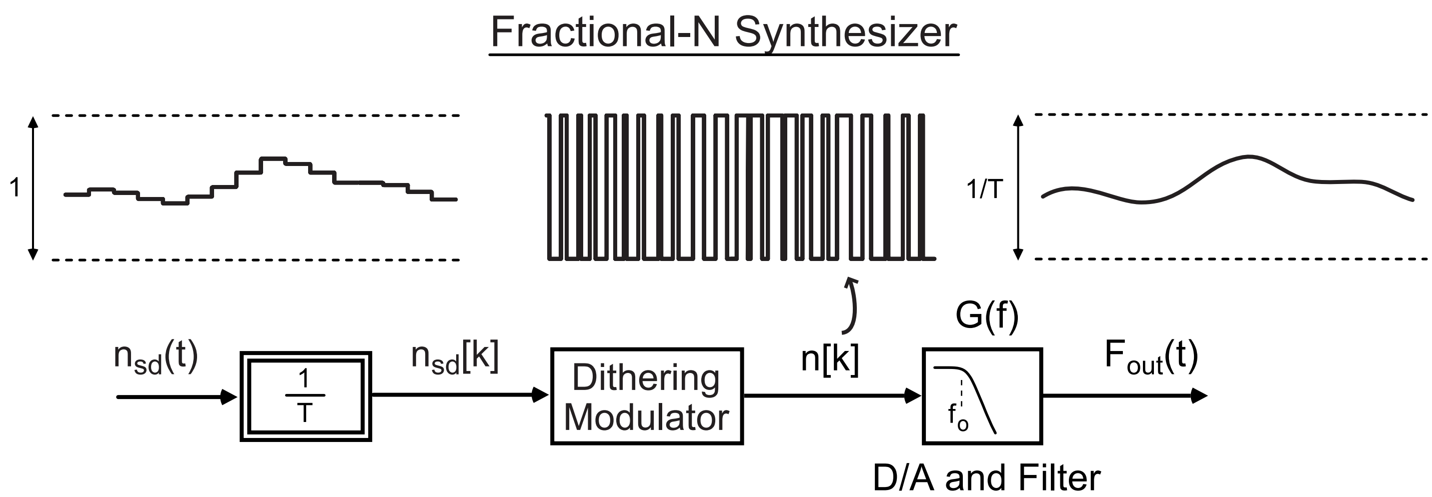
  - **Note that G(f) = 1 at DC**

Now Consider Impact of Divide Value Variations



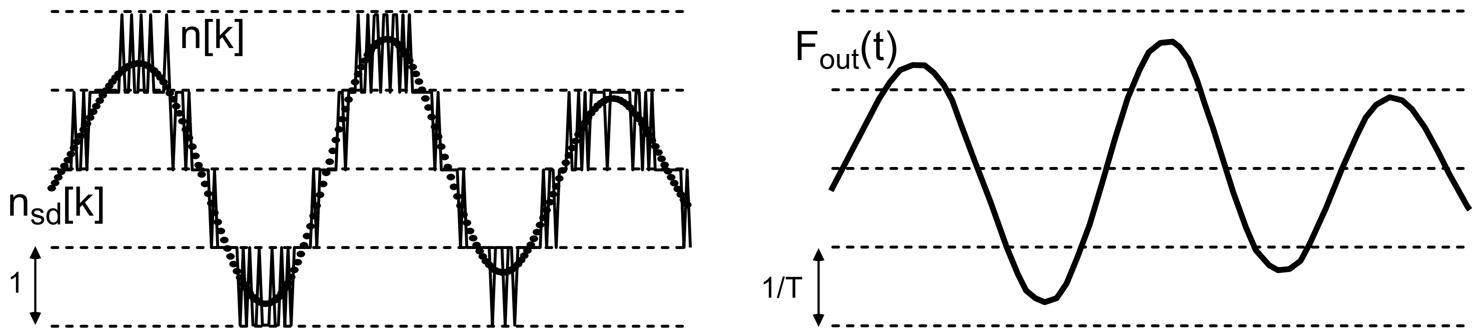
Divider Impact For Classical Vs Fractional-N Approaches

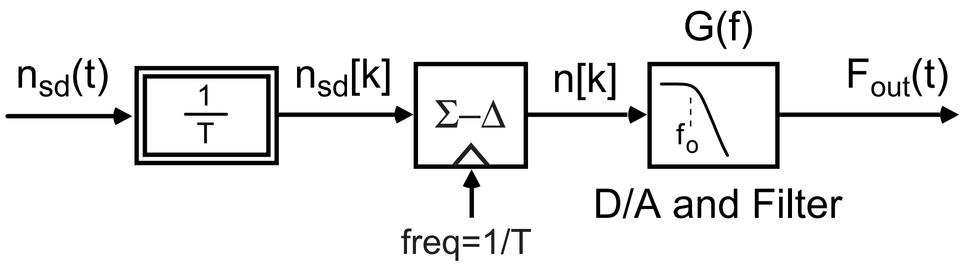




- **Note: 1/T block represents sampler (to go from CT to DT)**

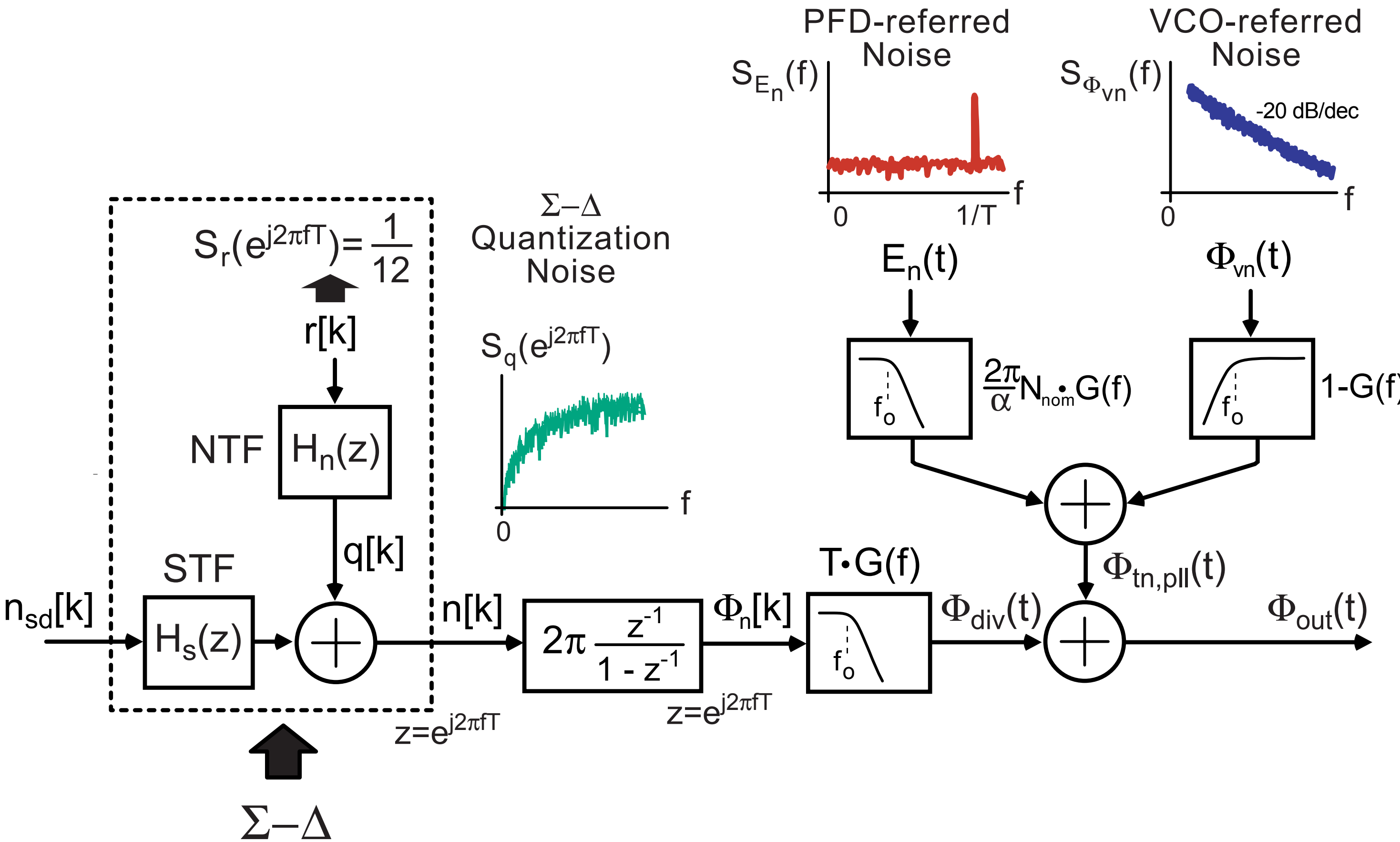
Focus on Sigma-Delta Frequency Synthesizer





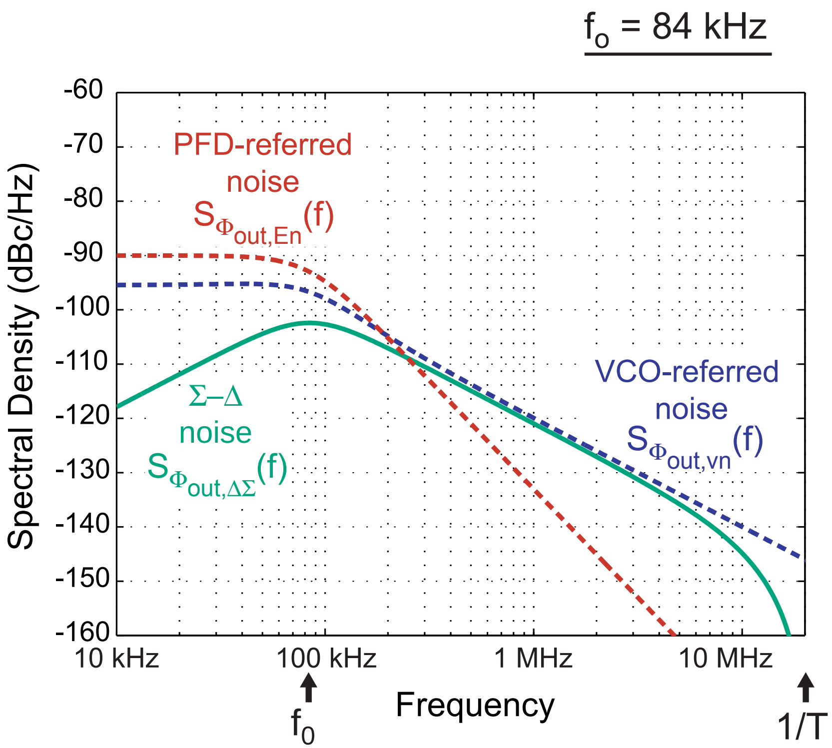
- **Divide value can take on fractional values**
  - **Virtually arbitrary resolution is possible**
- **PLL dynamics act like lowpass filter to remove much of the quantization noise**

Quantifying the Quantization Noise Impact



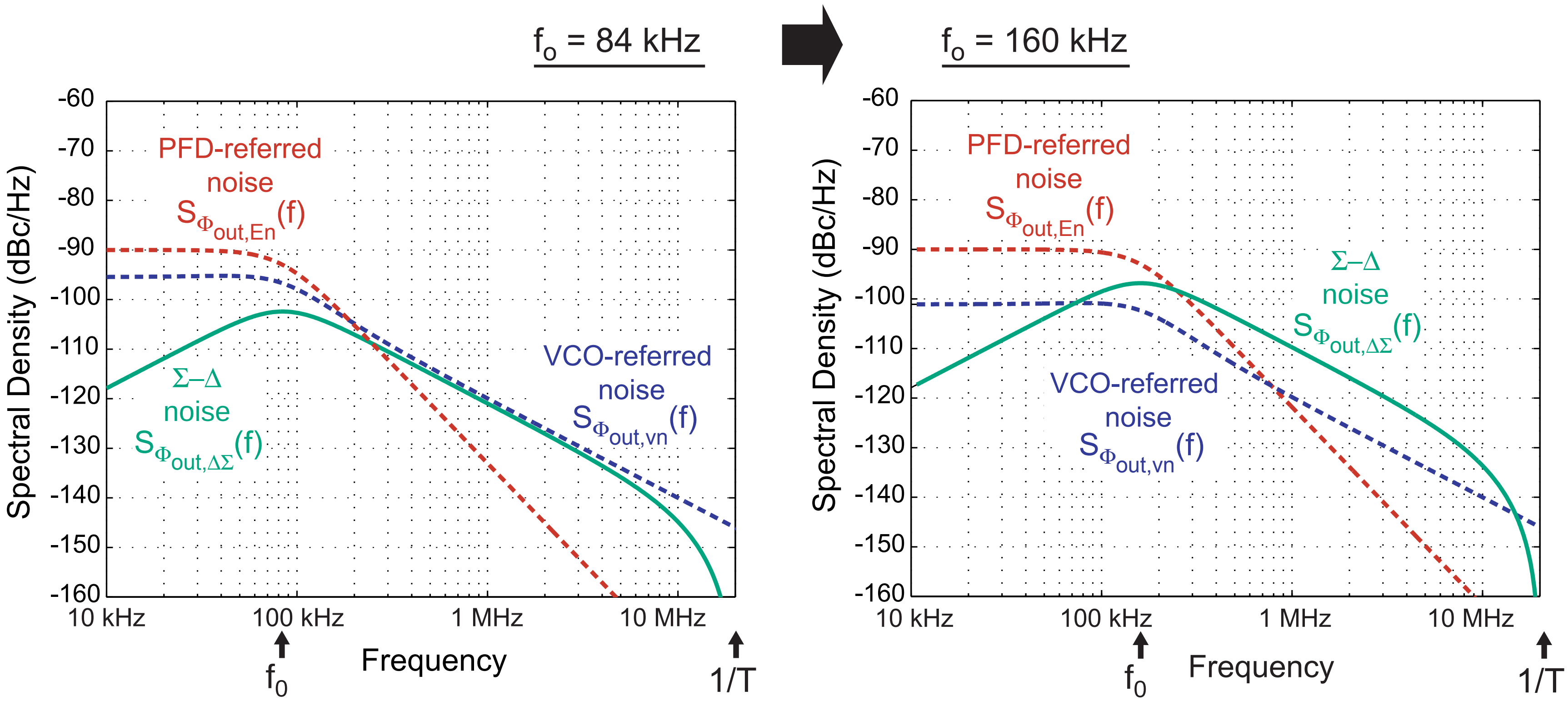
- **Calculate by simply attaching Sigma-Delta model**
  - **We see that quantization noise is integrated and then lowpass filtered before impacting PLL output**

A Well Designed Sigma-Delta Synthesizer



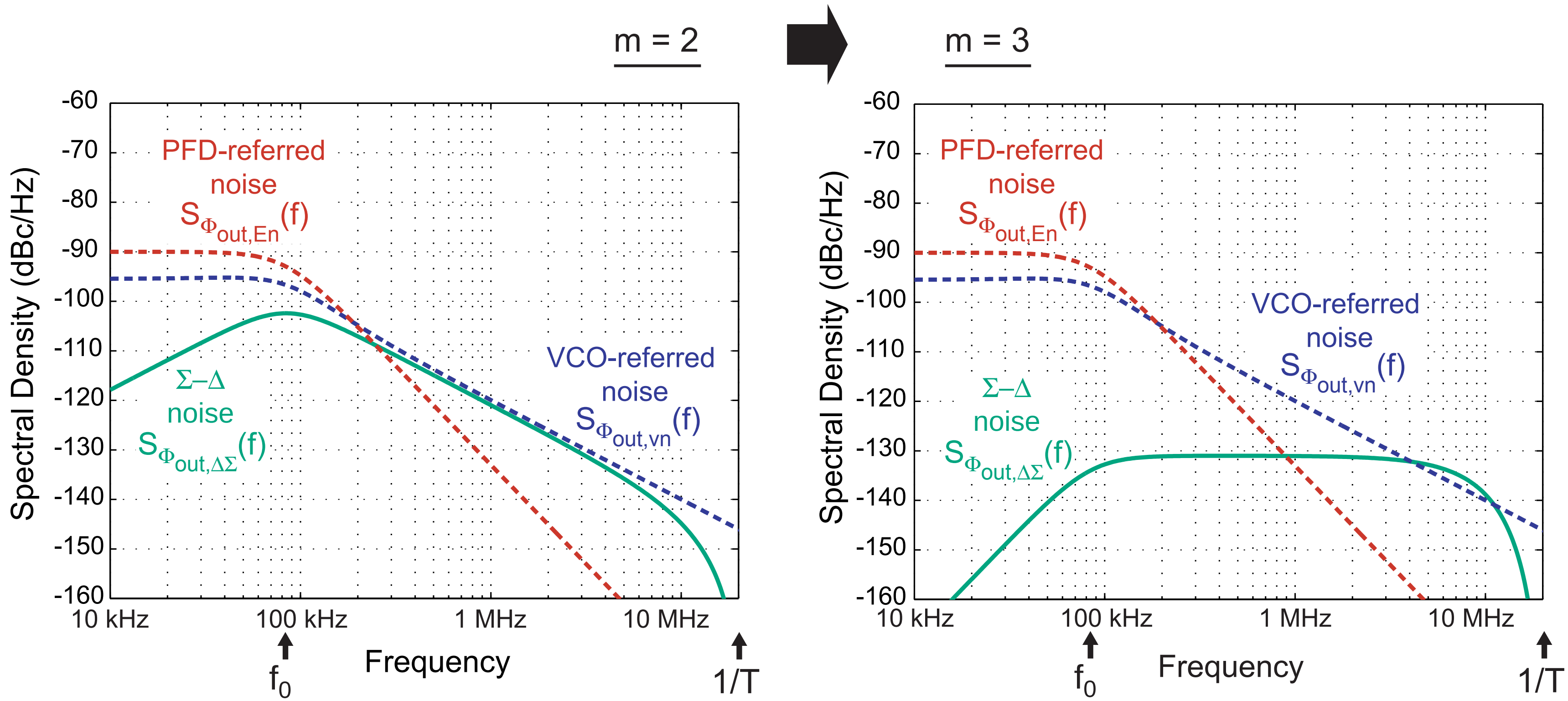
- **Order of G(f) is set to equal to the Sigma-Delta order**
  - **Sigma-Delta noise falls at -20 dB/dec above G(f) bandwidth**
- **Bandwidth of G(f) is set low enough such that synthesizer noise is dominated by intrinsic PFD and VCO noise**

Impact of Increased PLL Bandwidth



- **Allows more PFD noise to pass through**
- **Allows more Sigma-Delta noise to pass through**
- **Increases suppression of VCO noise**

Impact of Increased Sigma-Delta Order



- PFD and VCO noise unaffected
- Sigma-Delta noise no longer attenuated by G(f) such that a -20 dB/dec slope is achieved above its bandwidth